SISTEMA E-HEALTH DE ADQUISICIÓN Y ALMACENAMIENTO DE VARIABLES FISIOLÓGICAS, OBTENIDAS DE DISPOSITIVOS COMERCIALES, NECESARIAS PARA PREDECIR EL NIVEL DE INSULINA

Integrante

Giovanni Sabogal Cespedes

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ D.C.
2020

SISTEMA E-HEALTH DE ADQUISICIÓN Y ALMACENAMIENTO DE VARIABLES FISIOLÓGICAS, OBTENIDAS DE DISPOSITIVOS COMERCIALES, NECESARIAS PARA PREDECIR EL NIVEL DE INSULINA

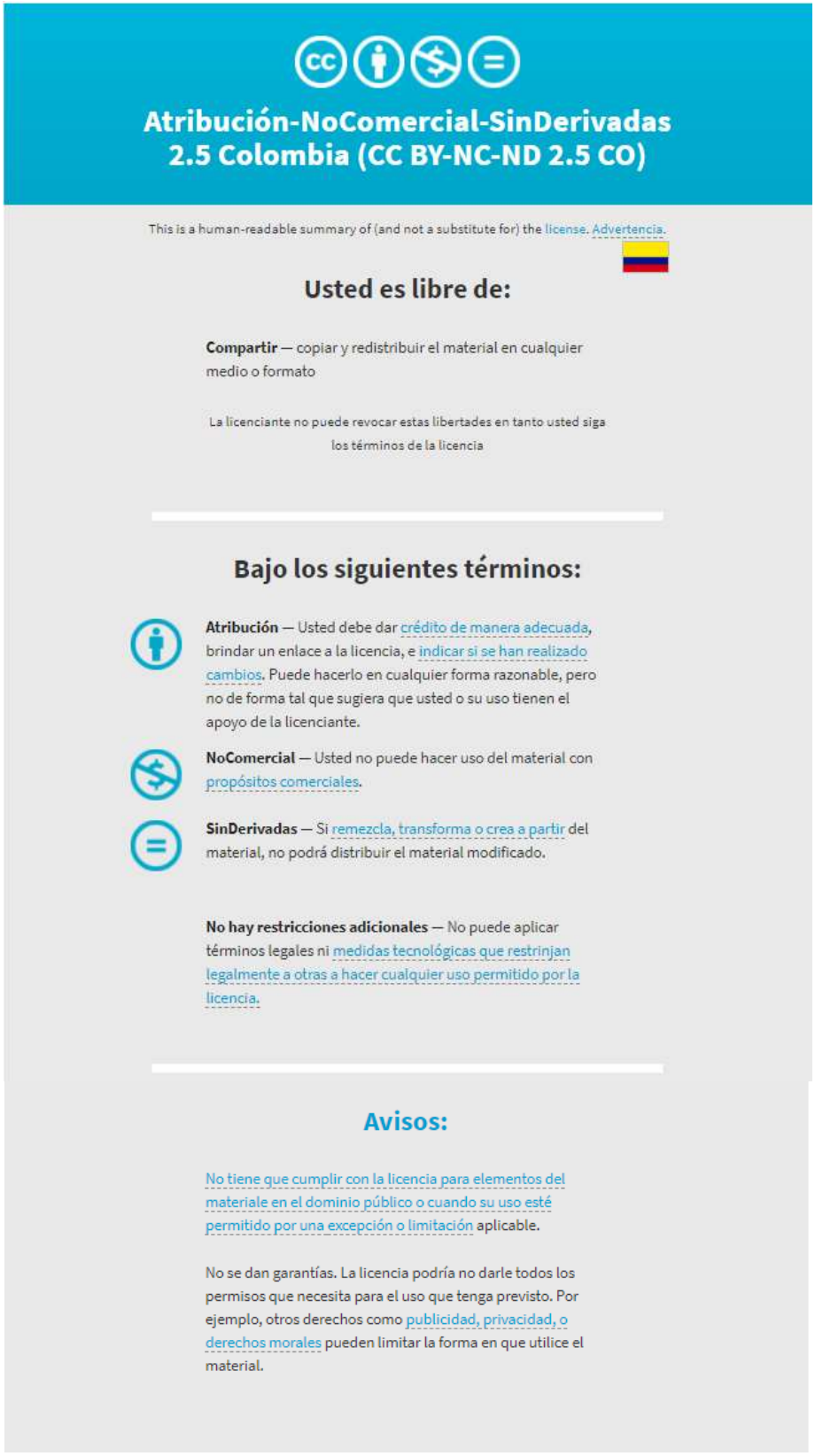
Integrantes
Giovanni Sabogal Cespedes

Trabajo de tesis

Director

PhD. Yuri Andrea Jiménez

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ
2020



Atribución-NoComercial-SinDerivadas 2.5 Colombia (CC BY-NC-ND 2.5 CO)

This is a human-readable summary of (and not a substitute for) the license. Advertencia.

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:

Atribución — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.

NoComercial — Usted no puede hacer uso del material con propósitos comerciales.

SinDerivadas — Si remezcla, transforma o crea a partir del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del materiale en el dominio público o cuando su uso esté permitido por una excepción o limitación aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como publicidad, privacidad, o derechos morales pueden limitar la forma en que utilice el material.

Nota de aceptación:

_____

_____

_____

_____

_____

_____

_____

_____
Firma del presidente del jurado

_____
Firma del jurado

_____
Firma del jurado

PAGINA DEDICATORIA

Este trabajo se lo dedico a mis padres que desde pequeño me inculcaron y apoyaron la disciplina de estudiar y luchar por los sueños y una vida mejor.

A mis hermanos como motivación para que ellos logren también realizar cada una de sus metas.

A la Dra. Yury Andrea Jiménez, porque sin ella no hubiera logrado culminar este trabajo de grado.

A mi pareja Valentina Mondragón quien creyó en mi desde el principio y es quien logra sacar lo mejor de mí en cada instante.

AGRADECIMIENTOS

El presente trabajo de tesis primeramente me gustaría agradecer a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Segundo a mi familia por su apoyo incondicional, por su paciencia, por el amor y consejos que me brindan, en especial a mi padre John Jairo Sabogal Guzmán y a mi Madre Sandra Edelmira Cespedes Perdomo.

A mi directora de tesis, Dra. Yury Andrea Jiménez por su esfuerzo y dedicación, quien, con sus conocimientos, su experiencia, su paciencia, su motivación, consejos y su gran forma de ser, como también su gran apoyo en esta época tan difícil que vive el país, ha logrado en mí que pueda terminar mis estudios con éxito.

A el compañero de Ingeniería en Sistemas y computo, Javier Sarmiento, quien desinteresadamente me apoyo como tutor para poder comprender conceptos y funcionalidad del campo de la ingeniería de sistemas, necesarios para el desarrollo de esta tesis.

Mi agradecimiento a Valentina Mondragón Restrepo, por su apoyo incondicional, los consejos, el gran contenido como persona que compartes conmigo, la motivación a siempre se mejor y sobre todo por todo el cariño que has expresado por a lo largo de estos años.

A mi tío Luis Fernando y a su familia, porque todos fueron un gran apoyo y fuente de motivación en el proceso, seguramente sin su apoyo posiblemente habría que hacer varias pausas durante mi carrera, pero no fue así gracias a mi Luis y a la familia que siempre nos apoyó y lo apoyo en sus decisiones, le deseo muchas bendiciones a toda la familia, de mi parte estaré eternamente agradecido

Al ingeniero Iñaqui por toda la colaboración en el proceso realizado en este trabajo de grado ya se hizo un trabajo en conjunto con su grupo de investigación para poder brindar algo positivo al mundo, gracias por ser esa gran persona que pone un muy buen ejemplo de cómo puede ser esa vida académica en el exterior.

A Yina y Juan quienes actualmente quien ha mostrado su apoyo no solo hacia mí, sino hacia mis hermanos también desde un inicio.

También me gustaría agradecer a mis profesores durante toda mi carrera profesional porque todos han aportado con un granito de arena a mi formación tanto en el campo de la ingeniería de Electrónica, como también la ingeniería en Sistemas.

CONTENIDO

LISTADO DE TABLAS

LISTADO DE IMÁGENES

LISTADO DE ANEXOS

GLOSARIO

ARDUINO: es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

SERVIDOR: es una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, llegando a tener computadoras dedicadas a las cuales se les conoce como “servidores”.

APP WEB: son aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

DIABETES: es una enfermedad crónica e irreversible del metabolismo en la que se produce un exceso de glucosa o azúcar en la sangre y en la orina; es debida a una disminución de la secreción de la hormona insulina o a una deficiencia de su acción.

BASCULA: es un Instrumento para medir pesos, que consiste en una plataforma donde se coloca lo que se quiere pesar, un sistema de palancas que transforma la variación de las propiedades de un material en función del peso, y un indicador que marca el peso.

Palabras clave:

Freestyle libre, Fitbit Ionic, Bascula, sistema de monitoreo remoto, aplicación web, celda de carga, variables fisiológicas, diabetes, DM.

1. INTRODUCCIÓN

La Organización Mundial de la Salud (OMS) define la diabetes como una enfermedad crónica, consecuencia del mal funcionamiento del páncreas al dejar de producir insulina, también producida cuando el organismo es resistente a la insulina que produce el páncreas, pues la insulina es la hormona encargada de regular el azúcar en la sangre. Existen tres tipos de diabetes principales: diabetes tipo 1, diabetes tipo 2 y diabetes gestacional. En Colombia se cuenta con un índice de prevalencia del 8.36 de personas diagnosticadas con diabetes, con un número de casos aproximado de 2.206.440 personas entre 20 y 79 años [1].

Si la diabetes no es controlada, puede causar que los pacientes sufran de hiperglucemia, que con el tiempo genera daños graves en varios órganos (ojos, cerebro, corazón), puede afectar las extremidades de tal forma que podría hacer que el paciente sufra alguna amputación, en especial el sistema nervioso y los vasos sanguíneos [2]. La diabetes es una enfermedad degenerativa y a su vez afecta el metabolismo generando así el Síndrome Metabólico (SM), aumentando a su vez la probabilidad de unas complicaciones cardiovasculares, también nefropatía y neuropatía [3].

Los planes de tratamiento para un paciente diabético tienen como objetivos: controlar los síntomas presentes, prevenir o retrasar la aparición de las complicaciones asociadas, permitirle al paciente que lleve una vida completa y activa, y hacer que el paciente cuide de su enfermedad bajo una guía profesional adecuada. Para llevar a cabo lo anterior es sumamente necesario llevar un control de la glucemia. Se consigue mediante algunas estrategias dependiendo del tipo de diabetes. En el caso de los diabéticos tipo 1 es prioritario el suministro de insulina (ya que esta persona es insulinodependiente), así como también se debe utilizar inmunosupresores para reducir la insulitis. Para el caso de los diabéticos tipo 2, pueden seguirse diversos procedimientos según la etimología del trastorno. La dieta y el ejercicio junto con hipoglucemiantes orales (es un conjunto de drogas que se caracteriza por producir una disminución de los niveles de glucosa luego de su administración vía oral) son los tratamientos más habituales, pero no se descarta la administración de insulina [4].

---

[1] Federation International Diabetes, “Peaje mundial de diabetes,”. {En línea}. 2019 {01 marzo de 2020}. disponible en: (https://www.diabetesatlas.org/en/sections/worldwide-toll-of-diabetes.html.)

[2] “Diabetes” {En línea} página OMS, 2018. {26 marzo de 2020}. disponible en: (https://www.who.int/es/news-room/fact-sheets/detail/diabetes).

[3] Comportamiento de variables clínicas, antropométricas y de laboratorio en pacientes con síndrome metabólico. {En línea}. {07 mayo de 2020}. disponible en: (http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1727-897X2011000200004).

[4] FAUS, SANCHEZ POZO. Tratamiento, control y seguimiento fármacoterapéutico del paciente. En: Rev. Pharm Care Esp. No. 3, (2001) p. 242.

En la dieta, el monitoreo y mantenimiento de un peso aceptable es fundamental para un paciente diabético, es así que la cantidad de calorías habrá que decidirla y monitorearla con el objetivo de alcanzar y monitorear el peso ideal. Por otra parte, en el tratamiento para una persona diabética, es necesario llevar un control estricto de la presión arterial por debajo de los 130/85 mm/ Hg ya que disminuye el riesgo de retinopatía, nefropatía, efectos comunes que se desarrollan con el paso del tiempo en una persona diabética [5].

Gracias al uso de la tecnología y a las comunicaciones (TIC) surge un apoyo dirigido hacia el campo de la salud, incluyendo servicios de diagnósticos, tratamiento, prevención y también permite realizar un seguimiento a los pacientes de forma remota, todo esto con el fin de reducir costos en el sistema de salud. Es este tipo de prácticas basadas en tecnología de la información más las comunicaciones y medicina se conocen como E-Health o E-salud (salud electrónica en español).

Una de las ventajas que ofrecen las prácticas como E-Health es que brinda la oportunidad de ampliar el área en que se prestan servicios de salud a poblaciones con dificultades para el traslado continuo a sus centros de salud (personas de acceso remoto) [6].

Este trabajo de grado plantea la implementación de sistema que permite la adquisición y almacenamiento de un conjunto de variables (Frecuencia cardiaca, nivel de glucosa, cantidad de paso, cantidad de calorías y el peso) de dispositivos comerciales para determinar el nivel de insulina que el paciente diabético requiere en una sola base de datos.

Este trabajo de grado se realiza en conjunto con la investigación en desarrollo de algoritmos de predicción de insulina de la Universidad Complutense de Madrid, en el grupo de investigación, requiere la actualización de su base de datos MySQL con todas las variables fisiológicas pertinentes al campo de la diabetes (nivel de glucosa, frecuencia cardiaca, cantidad de pasos, cantidad de calorías, peso entre otras), de tal forma que se pueda actualizar los dato en tiempo real o en lapsos de 5 minutos aproximadamente hablando específicamente del Fitbit y del sensor Freestyle. Ya que en la actualidad su proceso de almacenamiento es por medio de ficheros que extraen a través del sensor, pero esto se debe realizar a mano por el usuario cada vez que se desea actualizar los datos en la base de datos.

Los investigadores de la Universidad Complutense de Madrid están utilizando dispositivos comerciales como Fitbit y Freestyle para la adquisición de los valores históricos de estos dispositivos. Para este trabajo de grado, se seleccionan el

[5] MEDIAVILLA BRAVO, José Javier. Complicaciones de la diabetes mellitus. Diagnóstico y tratamiento. En: Rev. SEMERGEN. vol. 27, (2001); p. 137.
[6] J. A. Blaya, H. S. F. Fraser, and B. Holt. E-Health. Technologies Show Promise In Developing Countries. Health Aff., vol. 29. No. 2. (2010); p. 244.

sensor tipo parche Freestyle y el reloj Fitbit, teniendo en cuenta que estos dispositivos envían los datos a plataforma de los fabricantes de estos dispositivos, en las cuales no les permiten manipular o extraer la información, tan solo su visualización y se suben en distintas plataformas. Es por esta razón que se decide crear un sistema que permita la adquisición de las variables que provienen de los dispositivos comerciales (Fitbit, Freestyle y una báscula) en una sola base de datos.

La estructura de este documento está compuesta inicialmente por los antecedentes en donde indica brevemente los estudios hallados en el control y tratamiento de la diabetes, continuando con la justificación en donde se explica el por qué es importante llevar un control en el tratamiento del paciente diabético. Posteriormente se encuentra el planteamiento del problema que se abordará en este trabajo de grado, para poder continuar con la definición de los conceptos más relevantes dentro del desarrollo del trabajo de grado. Al final se define el objetivo general y los específicos los cuales serán el eje central del desarrollo, y se culmina con las actividades que se llevarán a cabo para cumplir con las metas propuestas por los objetivos planteados.

2. GENERALIDADES.

2.1 ANTECEDENTES

En el estudio presentado en la revista española endocrinología pediátrica en el 2019 volumen 10 edición 2, se presenta un estudio en el cual se evalúa la calidad de vida de personas que padecen diabetes mellitus tipo 1 (DM1) y su relación con Prueba de hemoglobina A1c (HbA1c), en el que concluyeron que era menor la calidad de vida de los padres con respecto a sus hijos, y que un mejor control metabólico (medido por el valor de HbA1c) contribuye a una mejor percepción de calidad de vida [7].

2.1.1 EN ENFERMERÍA

En el estudio sobre el manejo de los dispositivos de insulina e información al paciente realizado en 2014 en la disciplina enfermería por Andrea Orcos Sánchez de la universidad de la Rioja, con el objetivo de diseñar una estrategia de educación sanitaria al paciente diabético insulina dependiente para conseguir una óptima administración de insulina y mejora en la cantidad de cuidados mediante un plan individualizado según las características fisiológicas del paciente. Para ello, organizan y entregan una serie de conocimientos sobre las buenas prácticas de hábitos saludables, el manejo correcto de dispositivos de insulina y el manejo efectivo del régimen terapéutico, concluyendo su trabajo con una guía compuesta por un lenguaje coloquial y sencillo que le permitirá a cada paciente alcanzar los conocimientos idóneos [8].

2.1.2 ALGORITMOS PARA CONTROLADORES APLICADOS AL MODELO DE GLUCOSA-INSULINA

Dayan Calupiña M. y Andrea García V. proponen el diseño, simulación y comparación de controladores de suministro de insulina clásicos y avanzados, aplicados al modelo Glucosa-insulina en el sistema de páncreas artificial para pacientes con diabetes Tipo 1. Realizando pruebas con simulaciones computacionales, iniciando con el modelo de glucosa-insulina de Hovorka. Se

---

[7] LACÁMARA ORMAECHEA, Nerea, BALSEIRO CAMPOAMOR, Marina, RUIZ SERRANO, Aranzazu, ROYUELA, Ana y MARTINEZ BADAS, Itziar. Relacion entre calidad de vida y control metabólico, tipos de tratamiento con insulina y monitorización de glucemia en diabetes mellitus tipo 1. Volumen 10. 2da edición. España: 2019 p. 64.

[8] SAINZ de ROZAS, APARICIO JARA, Gallardo and ACADÉMICO, “Manejo de los dispositivos de insulina. Información al paciente” 2013, p 65. Trabajo de investigación (Grado en Enfermería) Escuela Universitaria de Enfermería.

obtuvieron aproximaciones de primer orden, para después realizar los diseños de los controladores compuestos y estos son: PID, PID No Lineal y SMC Dinámico, a los que les agrega un feedforward y con ello se llega al diseño del controlador por matriz dinámica (DMC)[9], aclarando que los controladores son un circuito cerrado y es utilizado para la implementación del páncreas artificial, así como se muestra en la Imagen 1.

Imagen 1. Circuito cerrado utilizado para implementación de páncreas artificial.

Fuente: Dayan J. Calupiña M. y Andrea P García V

En la investigación hecha en 2017 por Pamela A, Néstor S, Ethel C, en donde se registra la evolución de los sistemas de monitoreo y control de glucemia presente en el paciente con DM. Empezando por analizar la bomba de insulina hasta llegar al páncreas artificial con el fin de dejar de forma explícita los distintos dispositivos y tecnologías comerciales disponibles en la actualidad. Así como las ventajas y desventajas que tiene los sistemas de monitoreo y control de glucemia, en donde se encuentra la tecnología como la de infusión continuo subcutáneo de insulina (ICSI) tipo “Parche”, otra tecnología es el Monitor continuo de Glucosa (MCG). Estas dos tecnologías se unieron para tener un sistema complementario llamado SAPT, para después evolucionar a lo hoy conocido como Páncreas artificial (PA), siendo este último un infusor de insulina subcutáneo por catéter igual que infusor subcutáneo de insulina (ICSI). Esta última cuenta con una mayor integración con el MCG, generando un circuito cerrado. Llegando a la conclusión de que la tecnología está expandiendo las fronteras terapéuticas de los pacientes que usan insulinoterapia, y como efecto se tiene un mejor control metabólico con una disminución de riesgos por hemoglobina [10].

[9] CALUPIÑA MOYA, Dayan y GARCIA VÁSCONEZ, Andrea. Diseño, simulacion y comparacion de controladores clasicos y avanzados, aplicados al modelo de glocosa-insulina en el sistema de páncreas artificial para pacientes con diabetes tipo 1. Trabajo de investigación (Grado en ingeniero en electrónica y control) 2018, 125. Escuela politécnica nacional. Facultad de ingeniería eléctrica y electrónica.

[10] APABLAZA, Pamela, SOTO, Néstor y CODNER, Ethel. De la bomba de insulina y el monitoreo continuo de glucosa al pancreas artificial. EN Rev. Med. Chil. vol. 145, No. 5, (May 2017) pp. 630.

En la actualidad se encuentran también otros mecanismos de terapia y control de insulina que ayudan a regular los niveles de glucosa en el torrente sanguíneo. Los cuales se explicarán a continuación.

Iniciamos por el infusor subcutáneo de insulina (ICSI), también conocido como Bomba de insulina. El ICSI es un dispositivo electrónico desarrollado para suministrar insulina de forma continua al tejido subcutáneo, cuenta con un reservorio de insulina ultrarrápida que al ser entregado al tejido subcutáneo en cantidades pequeñas de forma continua permite una absorción rápida y estable. Este tipo de infusión reemplaza la secreción fisiológica basal del páncreas natural en el periodo del ayuno e interprandiales. Las dosis entregadas por el ICSI son programadas por el especialista cargo del paciente, siendo la programación en segmentos horarios acorde a los requerimientos del paciente en particular [11]. Las características de la terapia con infusor subcutáneo de insulina.

- Administración de insulina a través del catéter lo que hace posible el suministro de múltiples dosis de insulina evitando la necesidad de inyecciones.
- Suministro de insulina en dosis diferenciadas a lo largo del día.
- Los basales temporales permiten aumentar o disminuir de forma transitoria y suavizada la insulina basal para manejo de situaciones en las que se hace ejercicio o días en que el paciente se encuentre enfermo. Tiene la característica de cambiar la insulinoterapia programada en el ICSI en un porcentaje variable dentro de un rango acotado, durante un periodo de tiempo programado.
- Entrega de insulina pradial (bolos) con diferentes duraciones y ritmos, lo que permite la insulinoterapia diferenciada según la composición y duración de la absorción de los nutrientes.

A continuación, se mostrarán los tipos de infusores continuos subcutáneos de insulina (ICSI) en la Tabla 1.

[11] Ibid., p. 631.

Tabla 1. Tipos de infusores continuos subcutáneos de insulina (ICSI)

Tipo	Descripción	Modelo (Marca)
ICSI estándar	- Equipo con reservorio de insulina que aporta la hormona al paciente a través del set de infusión* - Sin monitor continuo de glucosa intersticial - Con o sin comunicación inalámbrica con glucómetro - Con o sin control remoto - Algunos modelos impermeables al agua	- Paradigm 515/715 (Medtronics®) - Accu-Chek Combo (Roche®) - Danna Diabecare IIS/R (Sooil®) - One Touch Ping (Animas®)** - T:flex (Tandem®)**
ICSI tipo "parche"**	- Reservorio de insulina desechable con la cánula incorporada que se adhiere a la piel como un parche sin mediar catéter-Control remoto comanda todas las funciones propias de un ICSI. Se comunica inalámbricamente con la parte que va adherida a la piel - Resistente al agua (no el control remoto) - Comunicación inalámbrica con glucómetro	- Omnipod (InsuletCorp®)**
"Sensor-augmented pump therapy" sin auto-suspensión automática de la infusión de insulina	- Dispositivo dual con ICSI más monitoreo continuo de glucosa en tiempo real (MCG-TR) - El equipo muestra en su pantalla la glicemia intersticial de forma continua y grafica la información de sus oscilaciones en el tiempo - Además de las alarmas programables propias de todo ICSI cuenta con alarmas programables para el MCG-TR - Requiere dos inserciones con aguja retráctil (no queda en el paciente): una para la cánula del set de infusión y otra para el sensor de glicemia - El sensor se comunica en forma inalámbrica al dispositivo - Al existir hipo o hiperglicemia no hay cambios en la administración de insulina, ya que infusor y monitor de glucosa no están comunicados para modificar la infusión	- Vibe (Animas®)** - T:Slim G4 (Tandem®)**
"Sensor augmented pump therapy" con auto-suspensión automática de la infusión de insulina frente a hipoglicemia	- Dispositivo dual con ICSI más MCG-TR - Con comunicación entre el infusor y el monitor de glucosa que permite la suspensión de infusión de insulina frente a hipoglicemia o antes que esta se produzca - Posterior a la suspensión la infusión de insulina se reinicia automáticamente - El protocolo de suspensión y reinicio de la infusión de insulina varía según el modelo de la bomba - No posee la capacidad de modificar automáticamente la infusión de insulina ante hiperglicemia	- Paradigm Veo (Medtronics®) - Minimed 640G (Medtronics®)
"Páncreas artificial y sistema híbrido"	- Equipo dual que tiene infusor de insulina y sensor de glucosa - Tiene un programa que integra el aporte de insulina con los niveles de glicemia intersticial - La infusión de insulina cambia ante hipoglicemia e hiperglicemia o según los niveles de glucosa intersticial - En páncreas artificial los bolos prandiales se entregan cuando el paciente informa que existirá una comida - En sistema híbrido el paciente necesita contar hidratos de carbono y utilizar asistente de bolos	- MiniMed 670G (Medtronics®), primer sistema híbrido con aprobación FDA en septiembre de 2016 - Múltiples sistemas en investigación

Fuente: APABLAZA, Pamela, SOTO, Néstor y CODNER, Ethel

La siguiente tecnología surgió una década después del ICSI, el monitoreo continuo de glucosa (MCG) permite medir el nivel de glucemia intersticial del tejido subcutáneo de forma continua, con intervalos de suministro de 5 min. Inicialmente era usado si era solicitado por el médico y su análisis era de forma retrospectiva. El MCG está diseñado para determinar la glucosa intersticial utilizando un sensor con glucosa oxidasa, la cual es una enzima que concentra la reacción electroquímica entre glucosa y O2, Como resultado se obtiene una corriente eléctrica en nano Amperios (nA), señal bastante pequeña que es enviada por un medio de transmisión inalámbrico desde el transmisor hasta el respectivo receptor del MCG. Esto es posible por medio de un algoritmo de calibración de corriente que entregada una relacionada con la información recopilada en *mg/dl* de la glicemia captada. Hay que

aclarar que el sensor de MCG es desechable y su vida es variable dependiendo del fabricante variando de 3 a 14 días, debiendo ser reemplazado el sensor por uno nuevo cuando caduca su funcionalidad[12].

El MCG mide glicemia intersticial a diferencia del glucómetro que mide la glucosa capilar. Y es precisamente por esta razón que existe un desfase entre estos dos, encontrándose un retardo fisiológico en el método de medición del MCG de 5 a 15 min. Debido a la demora al esparcir desde el capilar a través del intersticio, hasta el sensor ubicado a ese nivel, se presenta el retardo anteriormente mencionado. El desfase no tiene importancia si los niveles de glucemia se mantienen estables en el paciente, hallando una diferencia del 11% al 14% en los sensores modernos[13]. Los distintos tipos de MCG se observan a continuación en la Tabla 2.

Tabla 2. Tipos de monitor continuo de glucosa (MCG).

Tipo	Características	Modelo (Marca)
MCG profesional	- Lecturas de glicemias ciegas para el paciente - Utilizado como examen tipo "holter" de glicemia - La información se obtiene en forma retrospectivas una vez que es retirado el MCG - Programa computacional entrega la información como reportes que incluye gráficos, curvas y tablas	- Ipro (Medtronics®) - Freestyle libre Pro (Abbott®)*
MCG-TR o personal sin equipo infusor continuo de insulina	- De propiedad del paciente - Receptor del MCG con pantalla que muestra los niveles de glicemia intersticial y grafica su tendencia - Algunas marcas con transmisión de la señal a smartphone, que actúa como receptor - Alarmas programables para alertar de niveles o variaciones de glicemia potencialmente riesgosas para el paciente - No está integrado al ICSI - Se debe calibrar con niveles de glicemia capilar 2 a 4 veces al día	- G4 Platinum (Dexcom®) - G5 Mobile (Dexcom®)* - MiniMed Connect (Medtronics®)
MCG-TR o personal integrado con infusor continuo de insulina (SAPT)	- De propiedad del paciente - Equipo dual integrado con ICSI con pantalla que muestra los niveles de glicemia intersticial y grafica su tendencia - Puede tener o no sistema de suspensión automática de la infusión de insulina (Tabla 2) - Alarmas programables para alertar de niveles o variaciones de glicemia potencialmente riesgosas para el paciente - Se debe calibrar con niveles de glicemia capilar 2 a 4 veces al día	- Paradigm VEO (Medtronics®) - MiniMed 640G (Medtronics®) - Vibe (Animas®)* - T-slim G4 (Tandem®)*
MCG con sistema Flash	- Mide glicemias en tiempo real, pero no tiene receptor con pantalla que grafique todo el tiempo la información de la glicemia - La información es obtenida al pasar cerca del sensor el receptor, que funciona como lector tipo "scanner" - Sin alarmas programables	- Freestyle libre (Abbott®)

Fuente: Pamela Apablaza, Nestor Soto, Ethel Codner.

De los anteriores dos dispositivos antes mencionados (ICSI y MCG), se hizo la unión en un mismo dispositivo y dio lugar a una nueva tecnología llamada SAPT.

[12] Ibíd., p. 631.
[13] Ibíd., p. 632.

Actualmente los dispositivos integran la información de los niveles de glicemia intersticial y con un algoritmo computacional se realiza la infusión de insulina, el algoritmo utilizado es capaz detectar de forma automática la infusión de insulina frente a la hipoglicemia o antes de que ocurra, y reanuda la infusión posteriormente al estabilizarse [14].

La última tecnología en la que ha incurrido la medicina y por tanto aún continúa en investigación es el páncreas artificial (PA) y sistemas híbridos. El páncreas artificial también es conocido como biónico o en inglés closed loop, por medio de un catéter igual al ICSI aplica la insulina a las células subcutáneas, pero a diferencia del ICSI, cuenta con una mayor integración con el MCG, generando un circuito cerrado. Con lo anterior, se explica la forma en la que opera el páncreas artificial, aumentando y disminuyendo la entrega basal de insulina según lo reporta MCG [15].

En el caso de sistema híbrido, se integra en el dispositivo un sensor con el infusor ajustando en forma automática la entrega de infusión basal, con la condición de que el paciente debe continuar con la aplicación de los bolos, contando hidratos de carbono y midiendo glicemia capilar[16].

Rayman, Kroeger y Bolinder realizan un estudio utilizando el sensor FreeSttyle Libre Glucose Monitoring System en personas aspirantes a una licencia de conducción o conductores diabéticos de tipo 1 y tipo 2 tratados con insulina para poder conservar su licencia. Con el objetivo de determinar si las mediciones intersticiales (hechas con el sensor tipo parche no invasivo) de glucosa utilizando la tecnología flash de detección de glucosa, que pueden proporcionar información adicional para aumentar la conducción segura.

De los resultados se concluye que la información de la glucosa basada en sensores con flechas direccionales tiene el potencial de respaldar la evaluación de los niveles seguros de glucosa asociados con la conducción y ofrece distintas ventajas sobre las pruebas de glucosa en sangre para personas con diabetes tipo 1 y 2 para cumplir con los estándares de seguridad de conducción[17] (véase imagen 2).

---

[14] Ibíd., p. 632.

[15] Ibíd., p. 634.

[16] SERRANO, Valentina, LARREA MANTILLA, Laura, RODRÍGUEZ GUTIÉRREZ, René, SPENCER BONILLA, Gabriela, MALAGA, German, HARGRAVES, Ian y MONTORI, Víctor. Toma de decisiones compartidas en la atención de pacientes con diabetes mellitus: Un desafío para Latinoamérica. Rev. Med. Chil. Mayo, 2017, vol. 145, No. 5, pp. 642.

[17] Rayman, Kröger, and Bolinder. Could FreeStyle LibreTM sensor glucose data support decisions for safe driving. En: Rev. Diabet. Med. vol. 35, No. 4, (Abr 2018). pp. 491.

Imagen 2. Sensor FreeStyle Libre de Abbott.

Fuente: Inmaculada Rodríguez

En el trabajo de tesis de Inmaculada Rodríguez, Palomo realizó una revisión de las tecnologías invasivas, mínimamente invasivas y no invasivas que están siendo investigadas y desarrolladas actualmente con el fin de buscar una alternativa a los glucómetros a los glucómetros tradicionales (Los glucómetros a los que se refiere son a aquellos que necesiten de un pinchazo para obtener el valor de glucosa), ya que las multitudes de paciente diabéticos no realizan las mediciones diarias recomendadas debido al dolor y la molestia que se genera al hacer uso constante de estos glucómetros. Como resultado, la autora escoge la espectroscopia del infrarrojo cercano, como tecnología a desarrollar para la mejora de un sensor prototipo previo del grupo de ingeniería Biomédica de la Universidad de Sevillana[18].

En la investigación "Validation of the Fitbit One activity monitor device during treadmill walking" de Takacs Judit, Courtney L Polloc, Jerrad R Guenther, Mohammadreza Bahar, Chistopher Napier y Michael A. Hurt, estudian la validez y confiabilidad de un monitor de actividad, llamado FitBit One, en una población adulta con la característica que son personas sanas. Los autores no obtuvieron diferencias entre el conteo de pasos de FitBit y los conteos de los observadores, obteniendo resultados de (0.97 a 100). La confiabilidad entre dispositivos de conteo es buena, funcionando correctamente para todas las variaciones de velocidad al caminar.

Su porcentaje de error relativo es inferior al 1.3% La salida de distancia del monitor de actividad FitBit fue significativamente diferente de los valores de criterio para cada monitor a todas las velocidades y en se concluye con que los monitores de actividad FitBit One son dispositivos válidos y confiables para medir el recuento de

[18] RODRIGUEZ PALOMO, Inmaculada. Mejora del diseño de un prototipo de sensor no invasivo para la medida de glucosa en sangre. 2016, 134p. Trabajo de investigación (Trabajo de grado profesional Ingeniería Aeronáutica). Sevilla: Escuela Técnica Superior de Ingeniería Universidad de Sevilla. Departamento de Ingeniería de Sistemas y Automática.

pasos en poblaciones de adultos jóvenes sanos. Para finalizar se dice que la salida de distancia de los monitores es inexacta y debe tenerse en cuenta con precaución (véase imagen 3).

Imagen 3. Fitbit one.

Fuente: https://www.amazon.com/Fitbit-Wireless-Activity-Sleep-Tracker/dp/B0095PZHPE

2.2 PLANTEAMIENTO DEL PROBLEMA

La divulgación del término diabetes se atribuyó a Areteu de capadocia, de profesión, médico, del siglo II a.C. Areteu encontró los principales síntomas de la diabetes haciendo énfasis en su naturaleza progresiva y crónica. Y no fue sino hasta el siglo XVIII cuando un médico escocés llamado William Cullen, le otorgo el "apellido" a la diabetes, agregando al final del término mellitus, traducido del latín [19].

La diabetes a nivel mundial se ha convertido en uno de los más crecientes retos para la salud en el siglo XXI, en el 2019 el número de personas que padecen de diabetes se ha triplicado con respecto a unos 20 años atrás (2000), en donde la estimación global de los adultos con diabetes era para ese entonces de 151 millones. Para el 2009 había crecido en un 88% llegando a la suma de 285 millones para el 2019 se calculó que el 9.3% de los adultos de 20 a 79 años tienen diabetes, llegando así a una cifra alarmante de 463 millones de personas con diabetes[20]. Por otro lado, se encuentra que 1,1 millones de niños y adolescentes menores de 20 años presentan diabetes tipo 1 en el mundo [21].

[19] MALDONADO PAZ, Gerardo y DE LA CRUZ BERNARDA TÉLLEZ ALANÍS, María. Diabetes Mellitus Tipo 2 y sus efectos sobre procesos de Cognición Social. Cuernavaca, Morelos, 2019, 123p. Trabajo de investigación (grado de Doctor en Psicología). Universidad Autónoma.

[20] IDF Diabetes Atlas 9th edition 2019. {En línea}. {03 marzo de 2020}. 2019 disponible en: https://diabetesatlas.org/en/.

[21] S. N. Davis y LASTRA GONZALES, Guido. "Diabetes y Bajo Nivel de Glucosa (Hipoglicemia)". En: Rev. J. Clin. Endocrinol. Metab. vol. 93, No. 8, (Agos 2008.) p. 1.

Para el 2010 la proyección mundial de diabetes que se hizo para el 2052 era de 438 millones. Pasando cinco años después, en el 2015 esa predicción fue superada por 25 millones. La Federación Internacional de Diabetes (IDF) estima que habrá 578 millones de adultos con diabetes para 2030 y 7000 millones para 2045 aumentando 12 veces la cifra de personas diabéticas [22].

La revista El país en el año 2015 publicó un Ranking de los 10 países con el mayor número de casos identificados de Diabetes Mellitus (DM) a nivel mundial, de los cuales 3 países pertenecían al continente de América. Los tres países dentro del ranking eran EE.UU. (#3) con 29.3 millones de diagnósticos de DM, le seguía Brasil (#4) con 14.3 millones de diagnósticos y por último México (#6) con 11.5 millones de diagnosticados. De los tres países mencionados anteriormente, que pertenecían a América, Brasil es el país con la mayor cantidad de diagnósticos registrados en América del Sur [23].

Según las estadísticas brindadas por la IDF**,** la tasa de mortalidad a nivel mundial para el año 2019 fue de 4,2 millones de personas. Se encontró una relación existente entre los países con bajos y medianos ingresos y el índice alto de diagnósticos de diabetes, teniendo un 79% de adultos con esta patología [24].

En Latinoamérica (México, Centroamérica y América del sur) se diagnosticaron 40 millones de adultos que padecen de diabetes mellitus (DM), la mayoría de ellos de tipo 2. Se encontró que más del 80%de los pacientes con DM presenta una o más comorbilidades y cada una trae consigo su propio impacto sobre la salud y el trabajo terapéutico del paciente [25].

La IDF estimo que el 8,3% de prevalencia en personas diagnosticadas con DM en el mundo, esta entidad brinda los niveles de prevalencia de cada uno de los países en donde se encontró Colombia con un índice de intolerancia a la glucosa de 8,36 teniendo un número de casos de 2.206.440 siendo estas personas con edades entre 20 y 79 años [26].

---

[22] IDF. Número de víctimas de la diabetes {En línea}. {03 marzo de 2020} disponible en: https://diabetesatlas.org/en/sections/worldwide-toll-of-diabetes.html

[23] Revista EL País. Ranking de los 10 países con el mayor número de casos de DM 2015. {En línea}. {03 marzo de 2020} disponible en: https://elpais.com/hemeroteca/elpais/portadas/2015/.

[24] International Diabetes Federation. Cifras de estudios de Diabetes globales 2019. {En línea}. {03 marzo de 2020} disponible en: https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html.

[25] SERRANO, Valentina, LARREA MANTILLA, Laura, RODRÍGUEZ GUTIÉRREZ, René, SPENCER BONILLA, Gabriela, MALAGA, German, HARGRAVES, Ian y MONTORI, Víctor. Óp. Cit., p.641.

[26] VARGAS URICOECHEA, Hernando y CASAS FIGUEROA, Luz. Epidemiology of diabetes mellitus in South America: The experience of Colombia. En: Rev Clin. e Investig. en Arterioscler. vol. 28, No. 5, 2019 p. 246.

Con los anteriores datos se observó que son cercanos los datos a los del ministerio de salud en Colombia del año 2016 de 7.6% en hombres y 8.5% en mujeres, que representan dos millones de personas diagnosticadas y un millón sin serlo. Los factores de riesgo prevalentes para la Diabetes mellitus tipo 2 (DM2), están relacionados con el estilo de vida: según el índice de masa corporal (IMC) es la obesidad, la obesidad abdominal, la inactividad física y malos hábitos alimentarios, y otros factores que no son modificables, tales como, antecedentes familiares y la edad de la persona [27].

La diabetes mellitus es un problema que genera un mayor riesgo de complicaciones graves en las personas que padecen de esta enfermedad crónica, estando expuestos a riesgos tales como infarto de miocardio, ceguera, enfermedad vascular cerebral y amputación de los miembros pélvicos. Debido a la propia naturaleza la enfermedad, el paciente requiere de un autocuidado y de un proceso de control médico continuo para poder evitar las anteriores complicaciones mencionadas en un corto plazo y la reducción del riesgo a las complicaciones a largo plazo [28].

En la actualidad, el tratamiento médico que se les brinda a los pacientes con diabetes tipo 1 y 2, es la recomendación de un cambio de estilo de vida, de tal forma que disminuya factores en la persona como la obesidad. Promoviendo la actividad física y alterando la dieta, reduciendo así la incidencia de la diabetes mellitus tipo 2(DM2) [29]. El tratamiento está orientado a conseguir los siguientes aspectos:

- Desaparición de los síntomas relacionados con la hiperglucemia.
- Evitar la descompensación aguda de la enfermedad.
- En lo posible evitar o retrasar la progresión o aparición de las complicaciones crónicas divididas en Microangiopáticas, Macroangiopáticas.
    - ✔ Las complicaciones crónicas Microangiopáticas comprenden: retinopatía, nefropatía y neuropatía diabética.
    - ✔ Las complicaciones crónicas Microangiopáticas comprenden: (cardiopatía isquémica, enfermedad cerebrovascular, arteriopatía periférica.
- Reducir la tasa de mortalidad.

[27] RODRÍGUEZ y MENDOZA. Factores de riesgo de diabetes mellitus tipo 2 en población adulta. Barranquilla, Colombia. En: Rev. Colomb. Endocrinol. Diabetes Metab. vol. 6, No. 2, (May 2019); p. 87.

[28] ROMERO-VALENZUELA, Erika, ZONANA NACACH, Abraham y DE LOS ANGELES COLIN GARCIA, María. Control de glucosa en pacientes que asistieron al programa de educación DiabetIMSS en Tecate, Baja California. En: Med Int Méx. Vol 30, (2014) p 555.

[29] León, Isaura. Guías para el diagnóstico y tratamiento el síndrome metabólico y la diabetes tipo 2 (DM2). Nuevos criterios. En: Salus. vol. 8, No. 1. (2004) p. 4.

- En lo posible mantener la buena calidad de vida (haciendo que la persona pueda volver a los niveles considerados normales en la medicina para una persona promedio con niveles de glucosa estables).

Todo lo anterior, se realiza en la fase inicial del tratamiento contra la DM2, estableciendo un plan de actividad física y alimentación adecuados que permita controlar los niveles de glucemia. Desafortunadamente estas medidas son insuficientes en la mayoría de los pacientes y se debe llegar a establecer un tratamiento fármaco oral [30].

Al ser la diabetes tipo 2 (DM2) el resultado de la interacción cambiante y dinámica entre los defectos y la secreción y acción de la insulina, en la selección de la droga se tiene implícito las características del medicamento (modo de acción, costo y perfil de seguridad) y características del paciente como el: peso, edad, grado de hiperglucemia y comorbilidades.

Con lo anterior, se realiza en la fase inicial del tratamiento contra la DM2, estableciendo un plan de actividad física y alimentación adecuados que permita controlar los niveles de glucemia. Desafortunadamente estas medidas son insuficientes en la mayoría de los pacientes y se debe llegar a establecer un tratamiento fármaco oral. Es por ello por lo que ante la prevalencia de la DM que se presenta en la actualidad es necesario un mecanismo que permita realizar un monitoreo de los niveles de glucosa y otras variables que se requieran de manera práctica, que no solo sea una herramienta para las instituciones de la salud y el personal médico, para el tratamiento y diagnóstico de la enfermedad, sino que le permita al paciente tener autonomía y control sobre su salud.

Debido a esta situación, en este trabajo de grado se plantea el diseño e implementación de un sistema que permita la adquisición de un conjunto de variables necesarias para la predicción de la cantidad insulina requerida por el paciente diabético. Los valores obtenidos de las variables fisiológicas serán almacenados en un servidor (nube) usando una de las tecnologías disponibles en Internet de las cosas (IoT) para la transmisión de la información como GSM/GPRS o Wifi, entre otras. De acuerdo con lo anterior, se plantea la siguiente pregunta de investigación:

¿Cómo adquirir y transmitir los datos de las variables fisiológicas más relevantes, tomados de dispositivos comerciales, para la predicción del cálculo de insulina requerida por diabética?

[30] International Diabetes Federation - Facts & figures. Óp. Cit. p.1

2.3 OBJETIVOS

2.3.1 OBJETIVO GENERAL

Implementar un sistema de adquisición, procesamiento y almacenamiento de algunos datos fisiológicos relevantes para el cálculo de la predicción para el suministro de insulina en pacientes diabéticos.

2.3.2 OBJETIVOS ESPECÍFICOS

- Analizar los distintos métodos y mecanismos utilizados para realizar seguimiento a los niveles de insulina y otras posibles variables que afectan en la terapia de los pacientes con DM.
- Definir los requerimientos de medición de los niveles de insulina y otras posibles variables que se consideren necesarias para dar seguimiento al paciente utilizando las TIC.
- Diseñar el sistema de adquisición procesamiento y almacenamiento de variables fisiológicas definidas (módulos definidos).
- Implementar los módulos de monitoreo, procesamiento y visualización de la información.
- Validar el funcionamiento de cada módulo, así como integrarlos.

2.4 JUSTIFICACIÓN

La diabetes es una enfermedad crónica que se caracteriza por la existencia de niveles de glucosa altos, lo suficiente para considerarse peligrosos para el paciente, y generalmente esto está asociado a la falta de producción de insulina o resistencia de esta por parte del organismo [31]. Los programas de tratamiento intensivo de la diabetes se basan en la utilización de las múltiples estrategias (buena nutrición, ejercicio físico suficiente, farmacoterapia controlada y en constante actualización acorde a la evolución del paciente, monitorización de la glucemia, cambio de la conducta), esfuerzos y persistencia máximos para conseguir un control glucémico próximo a el ideal[32].

Según los expertos, dicen que el tratamiento para las personas diabéticas consiste en llegar a mantener en lo posible un nivel cercano de glucosa en la sangre una persona que no padece de diabetes. Las recomendaciones que se hacen para poder tener un buen control de la diabetes consisten en seguir un plan de comidas hecho por el nutricionista, realizar actividad física con regularidad, la aplicación oportuna y justa de la insulina (teniendo en cuenta las recomendaciones hechas por el médico encargado del control del paciente diabético) y llevar un control de medición de la glucosa en la sangre constantemente. Es por este motivo que se hace necesaria la existencia de un dispositivo que les permita medir el nivel de glucosa y ser almacenado **en la nube** el registro de forma tal que se pueda consultar remotamente[33].

De acuerdo con la necesidad expresada anteriormente, en este trabajo de grado se plantea una posible solución, con la implementación de un sistema capaz de adquirir un conjunto de variables que se consideren necesarias para el control del nivel de glucosa (estas posibles variables son Ritmo cardiaco, presión arterial, actividad del sueño y cantidad de pasos, nivel de glucosa) en el paciente, y almacenar esta información en un servidor. Con el objetivo de predecir la cantidad de insulina que requiere el paciente, validado por un médico que se encargará del control y evolución del tratamiento del nivel de glucosa de su paciente. En la Imagen 4 se muestran los valores recomendados de glucosa en el torrente sanguíneo. Si el nivel de glucemia está por encima el paciente es hipoglucémico generando efectos a largo plazo en los órganos, por el contrario, si está una persona por debajo (situación peligrosa cercana a 50) es una persona hipoglucémica, ambos casos

---

[31] LACÁMARA ORMAECHEA, Nerea, BALSEIRO CAMPOAMOR, Marina, RUIZ SERRANO, Aranzazu, ROYUELA, Ana y MARTINEZ BADAS, Itziar. Óp. Cit., p. 61.

[32] ANARTE, RUIZ DE ADANA, CARRERA, DOMIGUEZ LOPEZ, MACHADO, GONZALES MOLERO, CABALLERO, DE LA HIGUERA, GONZALES ROMERO, SANCHEZ, SORIGUER. Tratamiento de la diabetes mellitus (I). Av Diabetol., vol. 12, No. 18, pp. 1002.

[33] C. MARTÍN. Guía para personas con diabetes tipo 1 y 2. Rev. Natl. Inst. Diabetes Dig. Kidney Dis. vol. 2, (2013), p. 34.

requieren un mayor control por parte del paciente llenando la hoja con su registro de niveles de insulina, también requiere atención por parte de un profesional de la salud que haga la labor de seguimiento esta enfermedad.

Con lo anterior, se presenta la necesidad de un dispositivo que sea capaz de enviar al profesional de la salud las variables fisiológicas necesarias para la cantidad de insulina requerida por el paciente (con el objetivo de alcanzar los valores normales ver imagen 4) facilitando el control, así como liberando al paciente diabético de realizar el registro escrito de sus niveles de insulina y la necesidad de trasladarse forzosamente para llevar el registro a su médico encargado.

Imagen 4 . Niveles recomendados de glucosa.

Niveles recomendados de glucosa en la sangre para la mayoría de las personas con diabetes

Cuándo	Niveles recomendados	Mis niveles deseados
Antes de las comidas	de 70 a 130	de _____ a _____
1 a 2 horas después del comienzo de una comida	menor de 180	menor de _____

Fuente: Guía para personas con diabetes tipo 1 y 2.

2.5 MARCO REFERENCIAL

2.5.1 MARCO TEÓRICO.

2.5.1.1 Diabetes:

EL término que se utiliza es Diabetes Mellitus (DM), y describe un desorden metabólico en el que intervienen varios factores y se caracteriza por la hiperglucemia crónica con trastornos en el metabolismo de los carbohidratos, proteínas y grasa, causado por defectos en la secreción y/o en la acción de la insulina o de ambos [34].

Debido al desequilibrio metabólico continuo, a largo plazo genera complicaciones crónicas, esas complicaciones son las siguientes: la nefropatía diabética, causa más común de insuficiencia renal crónica terminal; retinopatía diabética segunda causa de ceguera en el mundo; neuropatía diabética que puede provocar úlceras, articulación de Charcot y ser causa de amputaciones en miembros inferiores.

2.5.1.2 Niveles de glucosa:

El término de glucemia se utiliza para describir la concentración de glucosa en la sangre y sus unidades son [*mg/dl*] o [*mmol/l*]. y se clasifican los niveles de concentración de la siguiente forma:

- Normoglucemia: Se caracteriza por que el nivel de glucosa se encuentra dentro de 70 hasta 70 [*mg/dl*][35], teniendo la excepción de que se puede subir hasta 180[*mg/dl*] después de ingerir alimentos, con la condición de que regrese a su nivel después de 1 o 2 horas[36].

- Hipoglucemia leve: se caracteriza por tener niveles de glucosa inferiores de 70[*mg/dl*][37].

- Hipoglucemia moderada: se caracteriza por tener niveles de glucosa inferiores a 55[*mg/dl*].

---

[34] RIVAS ALPIZAR, Elodia, ZERQUERA TRUJILLO, Gisela, Hernández Gutiérrez, Caridad y VICENTE SÁNCHEZ, Belkis. Manejo práctico del paciente con diabetes mellitus en la Atención Primaria de Salud. En: Finlay Rev. Enfermedades no Transm., vol. 1, No. 3, (2011) p. 230.

[35] BONDIA, VEHÍ, PALERM y HERRERO. El Páncreas Artificial: Control Automático de Infusión de Insulina en Diabetes Mellitus Tipo 1. En: Rev. Iberoam. Automática e Informática Ind. RIAI. vol. 7, No. 2, (2010), p. 7.

36 S. N. Davis y LASTRA GONZALES, Guido. Óp. cit., p E1.

[37] S. N. Davis y LASTRA GONZALES, Guido. Óp. cit., p E1.

- Hipoglucemia severa: se encuentra un nivel de glucosa menor a [*mg/dl*].

- Hiperglucemia: se presenta cuando los pacientes poseen niveles de glucosa superiores a 180[*mg/dl*].

2.5.1.3 Insulina:

Es una hormona que segregan las células beta dentro del páncreas, esta acción se realiza para controlar el nivel de glucosa presente en la sangre, y en menor grado otras sustancias dentro de los alimentos. La acción de la insulina es la que principalmente permite que se aprovechen bien los alimentos, es la responsable de que la glucosa, proteínas y grasas entren dentro de las células de los tejidos periféricos(músculos, hígado, entre otros) para ser utilizada[38].

La insulina es una hormona muy importante para el funcionamiento del cuerpo humano, y es por esta razón que continuamente hay una secreción basal con el objetivo de garantizar uno niveles mínimos de la hormona. Los niveles de insulina tienen un aumento después de cada comida, para poder aprovechar lo mejor posible los alimentos[39].

2.5.1.4 Acción de la insulina:

Se divide en tres etapas:

- Inicio de acción: consiste en el tiempo que tarda la insulina en llegar al torrente sanguíneo y para comenzar a reducir los niveles de glucosa presentes en la sangre.

- Acción máxima, o pico de acción: es el momento en que la insulina alcanza su máximo efecto en la reducción del nivel de glucosa en la sangre.

- La duración: es el tiempo que la insulina se mantiene reduciendo los niveles de glucosa.

---

[38]Cardona. Tratamiento de insulina. {En línea} Fundación para la diabetes, 2017. {28 marzo de 2020} disponible en: https://www.fundaciondiabetes.org/infantil/181/tratamiento-de-insulina-ninos.

[39] Cardona. Qué es la diabetes. {En línea} Fundación para la diabetes, 2017. {28 marzo de 2020} disponible en: https://www.fundaciondiabetes.org/infantil/176/que-es-la-diabetes-ninos#QueEsInsulina

Tipos de insulina:

- Insulinas de acción rápida (análogos rápidos): **Humalog, Novorapid** y **Apidra,** son análogos de acción rápida. Es un tipo de insulina, modificada molecularmente que permite variar el comienzo de su acción o duración. Las características de este tipo de insulina comienzan su acción entre 10 y 15 minutos después de su suministro. Su pico de actividad se encuentra entre los 30 y 90 minutos y su duración está en un aproximado entre 3 y 4 horas, es la más similar a la insulina producida por el páncreas de una persona sin diabetes[40].

- Insulina regular: Es también llamada insulina soluble o cristalina. La **Humulina Regular** y la **Actrapid** con ejemplos de estas, su acción comienza a los 30 o 60 minutos y el máximo de su actividad se encuentra entre las 2 y 3 horas y su duración ronda las 5 a 7 horas. Su uso ha decaído debido a la aparición de los análogos de acción rápida mencionados anteriormente, y su uso era principalmente para cubrir la carga de glucosa producida en el cuerpo, aunque su descripción de acción no se ajusta mucho a la forma de actuar de la insulina que es liberada por el páncreas.

- Insulina de acción intermedia (NPH): Se denomina insulina NPH (neutral Protamine Hagedorn) y Es utilizada como insulina basal, en el caso de la insulina humana se le ha añadido proteína llamada protamina con el objetivo de que la absorción sea más lenta, de esta forma la absorción tiene una duración entre 10 a 13 horas. Se encuentra su actividad máxima entre 4 a 7 horas después del suministro y su inicio de actividad a las 1 o 2 horas. Un aspecto importante de la insulina NPH es que es muy variable en cuanto a su actividad de un día para otro, por esta razón es necesario batirla durante unos momentos antes de su aplicación. Para utilizar la insulina como insulina basal, es necesario tres dosis al día.

- Insulinas de acción lenta o prolongada (análogos basales): Comercial mente se encuentra como Levemir y es un análogo de insulina con acción prolongada, esta insulina tiene aprobación para ser utilizada en niños mayor de 2 años y es menos variable que la insulina NPH y también que la Lantus. no posee un máximo de actividad sobresaliente, pero su duración se extiende hasta las 24 horas.

- La insulina Lantus, es un análogo de insulina que funciona también en niños mayores de 2 años, su duración también va hasta las 24 horas sin poseer un pico o máximo de actividad lo que hace que sea la candidata ideal como

[40] Cardona. Tipos de insulina {En línea} Fundación para la diabetes, 2017. {28 marzo de 2020} disponible en: https://www.fundaciondiabetes.org/infantil/181/tratamiento-de-insulina-ninos.

insulina basal, con un comienzo de actividad entre 1 y 2 horas, teniendo menor variabilidad que la insulina NPH.

- Insulinas combinadas: Son mezclas de insulina con distintas proporciones (entre análogos de acción rápida, regular o intermedia).

Los tipos de insulina disponibles en el mercado suelen clasificarse según su duración. Variando el tiempo de duración en ultrarrápida, rápida, intermedia y prolongada.

2.5.1.5 Ciclos glucosa insulina:

Los carbohidratos son la fuente de energía de los tejidos en el ser humano. Después de la alimentación, existe un incremento de glucosa en el torrente sanguíneo, en respuesta a esto, el páncreas secreta insulina, que es la hormona encargada de la regulación excedente de glucosa, y es debido a que permite el almacenamiento de la glucosa en el hígado y en los músculos. El estado contrario a lo anterior es en ayuno, que es cuando el nivel de glucemia desciende y para regular esa deficiencia de glucosa en la sangre, el mismo páncreas libera una hormona denominada glucagón[41].

2.5.1.6 Prueba de hemoglobina A1cN:

Según MedlinePlus, mide la cantidad de azúcar en la sangre(glucosa) adherida a hemoglobina. Definieron la hemoglobina como parte de los glóbulos rojos encargados de transportar el oxígeno de los pulmones al resto del cuerpo. La prueba HbA1c muestra el promedio de la cantidad de glucosa adherida a hemoglobina en los últimos tres meses. El periodo de tres meses es escogido debido al periodo de vida típico de un glóbulo rojo [42].

2.5.1.7 Presión arterial:

La presión arterial según Mediplus es la fuerza de la sangre al empujar contra las paredes de las arterias, por cada vez que el corazón late y bombea sangre hacia las arterias. La presión arterial aumenta cuando el corazón late, bombea sangre y es llamado presión sistólica, por el contrario, cuando el corazón está en reposo, entre latidos, la presión arterial baja y es llamado presión diastólica[43].

---

[41] CALUPIÑA MOYA, GARCIA VÁSCONEZ. Op. cit., p. 5.

[42] Prueba de hemoglobina A1c: Información en MedlinePlus sobre pruebas de laboratorio. {En línea}. {24 marzo de 2020} disponible en: https://medlineplus.gov/spanish/pruebas-de-laboratorio/prueba-de-hemoglobina-a1c/.

[43] MedlinePlus. Presión arterial alta. Medlineplus.gov. 2016 {En línea}. {01 abril de 2020} disponible en: https://medlineplus.gov/spanish/highbloodpressure.html.

2.5.1.8 Frecuencia cardíaca:

Se define como la cantidad de veces que el corazón late en un intervalo de tiempo estándar (1 minuto), a medida que la sangre es impulsada a través del sistema circulatorio, las arterias, se expanden y se contraen con el flujo sanguíneo. El pulso (frecuencia cardiaca) puede fluctuar o aumentar con el ejercicio, las enfermedades, las lesiones y emociones[44].

2.5.1.9 Niveles de referencia:

Según la definición brindada por la página del servicio médico de SANITAS, los valores que indican la presencia de hiperglucemia varían en función del momento en que se toma la muestra de glucemia. Si se realiza en ayunas (por lo menos 8 horas después de la última vez que se alimentó) los valores normales de glucosa se encuentran en el rango de 70 a 110 mg/dl de sangre. En los infantes el rango se encuentra entre los 40 y 100 mg/dl. Lo anterior se es conocido como glucemia basal[45].
La manera de averiguar si se presenta presión arterial alta es con un chequeo y los valores de una presión alta son; presión sistólica mayor de 140 o su presión diastólica es igual o 90 [46].

2.5.1.10 Internet de las cosas (IOT):

El internet de las cosas más conocido como IOT (Internet Of Things) por sus siglas en inglés, es un término que hace referencia a la posibilidad de conexión de distintos dispositivos a internet. Según la empresa de Cisco, el concepto de IOT nace aproximadamente en el año 2009, con el argumento de que fue en este año que el número de dispositivos conectados a internet superó a la población mundial, y desde entonces no ha hecho más que crecer el número de dispositivos conectados a internet.

Según el grupo de Cisco Internet Business Solutions Group (IBSG), estima que habrá cerca de 6 o 7 dispositivos conectados por usuario, lo que genera una estimación global de 50 mil millones de dispositivos conectados a la red en contraste a los 7.6 millones de personas a nivel mundial. Permitiendo nuevos tipos de redes, brindando nuevas oportunidades de investigación y aplicaciones prácticas en áreas

---

[44] Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure) - Health Encyclopedia - University of Rochester Medical Center. U of R. Universidad de Rochester: centro médico. {En línea}. 2016 {23 abril de 2020} disponible en: https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P03963.

[45] Qué valores de hiperglucemia determinan la existencia de un día. {En línea}. {25 marzo de 2020}. Disponible: https://www.sanitas.es/sanitas/seguros/es/particulares/biblioteca-de-salud/diabetes/valores-hiperglucemia.html.

[46] Qué valores de hiperglucemia determinan la existencia de un día. Óp. cit., p.1

como el medio ambiente, la educación, administración de recursos energéticos (ejemplo las luces en casa), salud pública (un ejemplo puede ser la Telemedicina), entre muchas otras más[47].

2.5.1.11 Servidores Para IoT:

- Google Cloud: Es un completo conjunto de herramientas para conectar, procesar, almacenar y analizar datos en la nube. La plataforma se compone de servicios en la nube escalables en la nube y totalmente gestionados. Permite obtener información en tiempo real, recopilando datos de los dispositivos permitiendo realizar análisis ad-hoc con google BigQuery de forma sencilla o la ejecución de análisis avanzados, con funciones de aprendizaje automático con Cloud Machine Learning Engine. Además, permite visualizar los resultados en paneles de control e informes detallados de Google Data Studio[48].

- Microsoft Azure: Al igual que Google Cloud es un conjunto de servicios en la nube y es la plataforma de computación en la nube publica de Microsoft y proporciona una alta gama de servicios in Could, incluidos los de procesamiento, analítica, redes y por supuesto almacenamiento. Cuenta con un periodo gratuito regalando $200 dólares, para gastar en los servicios. Todo esto dentro del periodo de prueba gratis, una vez ha concluido este periodo la facturación se hará de acuerdo con el uso de los servicios[49].

- Amazon web Services (AWS): es un servicio en la nube que proporciona capacidad informática en la nube de forma segura y con un tamaño medicable. Está diseñada con el objetivo de simplificar el uso de la informática en la nube a escala web para los desarrolladores. Proporciona un control completo sobre los recursos informáticos y le permite ejecutarse en el entorno informático que es probado por el propio Amazon[50].

---

[47] EVANS, Dave. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. 2011. 12p.

[48] Google Cloud IoT: servicios de Internet de las cosas totalmente gestionados. {En línea}. {04 mayo de 2020} disponible en: https://cloud.google.com/solutions/iot/.

[49] Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure. {En línea}. {04 mayo de 2020} disponible en: https://azure.microsoft.com/es-es/overview/what-is-azure/#most-popular-questions.

[50] AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube. {En línea}. {04 mayo de 2020} disponible en: https://aws.amazon.com/es/ec2/.

2.5.1.12 Telemedicina:

El gobierno de Chile define la telemedicina como “el suministro de servicios de atención sanitaria en los que la distancia constituye un factor crítico, realizado por profesionales que apelan a tecnologías de la información y de la comunicación con objeto de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención de salud y en actividades de investigación y evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven ”Según la OMS (1998)”[51].

2.5.1.13 E-Salud o E-Health:

En la investigación en salud y tecnologías de José Cepeda Diez, Xose Manuel Meijome, Azucena Santillán García definen el concepto de E-Health las aplicaciones de las nuevas tecnologías para los procesos que estén relacionados con la salud, realizando diagnósticos, controles, revisión de historias clínicas, prescripción digital, etc. Estando integrado con el avance del sistema sanitario de las TIC[52].

2.5.2 MARCO CONCEPTUAL

2.5.2.1 Bomba de insulina:

Consiste en un sistema de un tamaño muy reducido que permite infundir aplicar insulina a nivel subcutáneo de forma continua, tal suministro se hace con la intención de suplir las necesidades de insulina de una forma más natural (fisiológica) con el objetivo principal de conseguir un mejor control metabólico (HbA1c) y simultáneamente reducir el número y la cantidad del riesgo de hipoglucemia[53] (véase imagen 5).

[51] CEPEDA DÍEZ, Jose, MEIJOME SÁNCHEZ, Xosé, SANTILLÁN GARCÍA. Azucena. Innovaciones en salud y tecnologías: las cosas claras. En:Revista EfermeriaCyL, 1967. vol. 2, No. 5, (1967); p. 30.
[52] Ibíd., p.28
[53] LEVY, VIDAL y JANSÀ. Bombas de insulina. Una alternativa en el tratamiento de la DM1. En: Mesa redonda. Diabetes Mellitus. 2004. vol 60 No 2, (2004); p. 55.

Imagen 5. ICSI con bomba de insulina

Fuente: Dayan Jasmin Calupiña Moya, Andrea Patricia García Vásconez

2.5.2.2 Medidor continuo de glucosa (MCG):

Según la asociación de diabetes de Madrid, un medidor continuo de glucosa (MCG) son dispositivos que miden la glucosa de forma continua, ofreciendo lecturas de esta cada 5 minutos aproximadamente. Estos dispositivos están compuestos de un filamento flexible que es insertado de tal forma que quede debajo de la piel, teniendo una vida útil aproximadamente de 6 a 14 días, Varía dependiendo del fabricante. También contiene un transmisor que envía la señal a un dispositivo receptor que ofrece la visualización en pantalla de las lecturas[54].

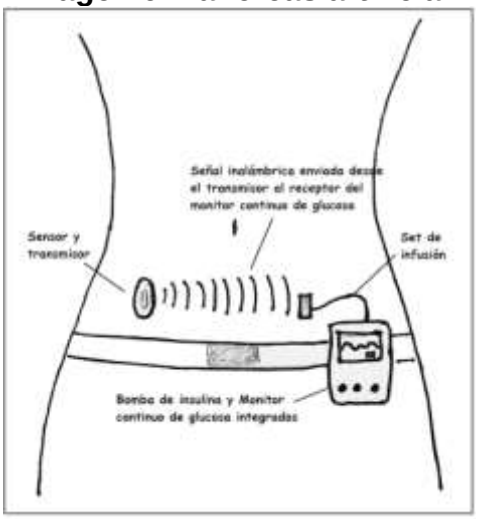
La diferencia entre este dispositivo y los medidores de glucosa capilar es que su objetivo a medir es la glucosa en el líquido intersticial.

2.5.2.3 Páncreas artificial:

El Páncreas artificial (PA) ver Imagen 6, también conocido como páncreas biónico, al igual que un ICSI, el páncreas suministra la insulina por medio de un catéter de forma subcutánea, teniendo este dispositivo un mayor acople con el MCG, generando un circuito cerrado permitiendo tener un mayor control al disminuir o aumentar la entrega basal de insulina[55].

[54] Medidores continuos de glucosa, ¿qué son? - Asociación Diabetes Madrid. {En línea}. {25 marzo de 2020} disponible en: https://diabetesmadrid.org/medidores-continuos-glucosa/.
[55] P. Apablaza, N. Soto, and E. Codner. Óp. cit., p. 634.

Imagen 6. Páncreas artificial.



Fuente: Apablaza Pamela, Soto Néstor, Codner Ethel

2.5.2.4 FitBit:

Es un dispositivo que monitorea Variables fisiológicas, contando con una monitorización continua del ritmo cardiaco, genera un registro de calorías quemadas a lo largo del día, lleva conteo de pasos al día, monitorización automática del sueño, mostrando el tiempo que pasas en las fases de sueño ligero, profundo y REM (Rapid eye movement)[56], Fitbit cuenta con varios modelos, uno de ellos se observa en la imagen 7.

Imagen 7. Dispositivo FitBit.

Fuente: https://www.amazon.com/-/es/inteligente-aluminio-incluidas-FB504GMBK-Original/dp/B07B48SQGT

[56]Fitbit Charge 3 | Pulsera de salud y actividad física avanzada. {En línea}. {03 mayo 2020} disponible en: https://www.fitbit.com/co/charge3.

2.5.2.5 FreeStyle Libre:

Es un sistema de monitoreo FLASH de glucosa, Freestyle Libre (ver Imagen 8) está diseñado para medir los niveles de glucosa del líquido intersticial en personas diabéticas de al menos 4 años tanto tipo 1 como tipo 2 por igual. El lector FreeStyle Libre utiliza una tecnología de sensor tipo parche no invasivo para el monitorio de los valores de glucosa, mostrándose instantáneamente de forma cómoda e intuitiva. Su sensor puede medir en intervalos mínimo de15 minutos entre una medición y otra los valores de glucosa y esta acción se puede realizar las veces que se desee durante el día, pero el parche dependiendo del fabricante de sebe cambiar cada 15 días aproximadamente[57].

Imagen 8. Dispositivo Freestyle medidor de glucosa que evita los pinchazos.

Fuente: https://www.freestyle.com.co/productos/freestyleLibre.html

2.5.2.6 Cloud Storage (CS):

Cloud Storage o Almacenamiento en la nube en español, es un tipo de servicio en *Cloud Computing* CC que permite la gestión de archivos o información en internet como si se tratase de un disco duro local, creando una serie de oportunidades[58] que a su vez permite que dispositivos con pocos recursos puedan almacenar su información, esto ha impulsado IoT y permite a su vez el avance en campos de investigación como telemedicina.

---

[57] FreeStyle Libre | Lector." {En línea}. {04 mayo de 2020}. Disponible en: https://www.freestyle.com.co/productos/freestyleLibre/lector.html.

[58] Memorias organizacionales en la era del almacenamiento en la nube." {En línea}. {04 mayo de 2020} disponible en: https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/6979/8659.

2.5.2.7 Microcontrolador:

Un microcontrolador se compone de tres bloques esencialmente: la CPU (Central Processing Unit), Las memorias (RAM: Memoria volátil y ROM: contiene las instrucciones que se deben realizar), las entradas y salidas. Los bloques se conectan entre ellos mediante un grupo de líneas eléctricas llamadas buses. La CPU, analógicamente es el "cerebro" del microcontrolador y actúa de tal forma tal que obedece las instrucciones almacenadas en la memoria. La CPU se encarga principalmente de extraer las instrucciones del programa desde la memoria, para poder interpretarlas (Traducción) y después ejecutarlas, este proceso principalmente se hace de dos formas dependiendo de la arquitectura que se utilice (arquitectura Harvard y Von Neumann). La CPU contiene circuitos para realizar operaciones aritméticas y lógicas elementales con los datos binarios, en la nombrada ALU(Arithmetic and Logic Unit) en español Unidad Aritmética y Lógica, un ejemplo de un microcontrolador se observa en la imagen 9 donde se ve un microcontrolador de la marca ST [59].

Imagen 9. Ejemplo de microcontrolador (STM32 Nucleo-64 boards (MB1136))

Fuente: https://electronicalibrexc.blogspot.com/2018/10/cual-es-la-diferencia-entre-stm32fx.html

2.5.2.8 Arduino:

Es un dispositivo Embebido que incorpora un microcontrolador re-programable con la característica de ser de código abierto, Basado en hardware y software flexibles y sencillo de utilizar. Consiste en un microcontrolador que es programado mediante un entorno de desarrollo basado en processing y una estructura de programación desarrollada a partir de Wiring[60].

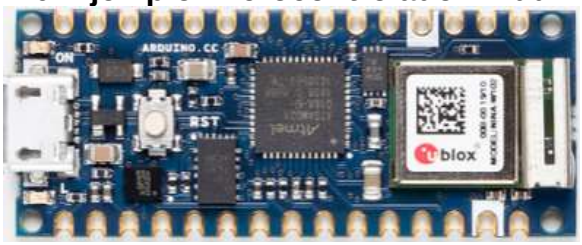
[59] VALDÉS PÉREZ, Fernando y PALLÁS ARENY, Ramón. Microcontroladores: fundamentos y aplicaciones con PIC. Marcombo Ediciones Tecnícas, 2007. 329p.

[60] ¿Qué es Arduino? Arduino.cl - Compra tu Arduino en Línea. {En línea}. {22 marzo de 2020} disponible en: http://arduino.cl/que-es-arduino/.

Fue desarrollado para facilitar la implementación de la electrónica en proyectos multidisciplinarios, ofreciendo beneficios prácticos para el desarrollo de proyectos y alguno de estos beneficios son:

- Entorno de desarrollo (IDE) sencillo llamado “Arduino IDE”.
- Es de código abierto.
- hardware y software extensible.

Imagen 10. Ejemplo microcontrolador Arduino Nano



Fuente: https://store.arduino.cc/usa/nano-33-ble-sense

2.5.2.9 GPRS:

Es el acrónimo que traduce (General Packet Radio Servicie), es una red basada en la conmutación de paquetes que funciona de forma paralela a la conmutación de circuitos de GSM[61]. Los principios de esta red son:

- Mantener los equipos de transmisión y la misma interfaz que GSM.

  Se realizan modificaciones pequeñas en la red GSM para poder transmitir datos a mayor velocidad.

2.5.2.10 Bluetooth:

Es un estándar desarrollado para la comunicación inalámbrica de datos de corto alcance. Sus características principales se pueden nombrar su robustez, consumo bajo, baja complejidad y bajo costo. Es una tecnología a pequeña escala y su banda de operación es 2.4GHz, y su señal tiene la capacidad de atravesar paredes[62].

---

[61] PRIETO DONATE, Francisco. GPRS (General Packet Radio Service). {En línea}. {25 de septiembre de 2018} disponible en: http://bibing.us.es/proyectos/abreproy/11372/fichero/Memoria%252F03+-+GPRS.pdf

[62] PRIETO DONATE, Francisc. 4. Tecnología Bluetooth 4.1. Introducción. {En línea}. {25 de septiembre de 2018} disponible en:

2.5.2.11 Wifi:

Es la tecnología que en la actualidad ofrece mayor la mayor cantidad de beneficios al costo más bajo entre todas las tecnologías inalámbricas. Es una tecnología económica y permite la interoperabilidad con equipos de diferentes fabricantes. Puede ser extendida para ofrecer funcionalidades mucho más allá de las planteadas originalmente por los fabricantes. Esto se debe la tecnología de Wifi utiliza estándares que son abiertos[63].

http://bibing.us.es/proyectos/abreproy/40048/fichero/VOLUMEN+1.+MEMORIA%252F4.+Tecnolog%C3%ADa+Bluetooth.pdf

[63] Introducción a las redes WiFi Materiales de entrenamiento para instructores de redes inalámbricas. International Centre for Theoretical Physics. {En línea}. 2012. {25 de septiembre de 2018} disponible en: https://docplayer.es/10897654-Introduccion-a-las-redes-wifi-materiales-de-entrenamiento-para-instructores-de-redes-inalambricas.html

2.5.3 METODOLOGÍA

Para este trabajo de grado se desea implementar un sistema de adquisición de variables fisiológicas de dispositivos comerciales para pacientes diabéticos, como se muestra en la Imagen 11 este sistema está compuesto por tres módulos adquisición, procesamiento y almacenamiento.

En el módulo de adquisición se obtendrán los datos de un conjunto de variables (por ejemplo: frecuencia cardiaca, nivel de glucosa, cantidad de pasos(colesterol), presión arterial, calidad del sueño). Posteriormente está la etapa de procesamiento de los datos obtenidos en la etapa anterior. A continuación, los datos serán almacenados en una base de datos.

El esquema que se muestra a continuación (ver imagen 11) representa el sistema que se desea implementar.

Imagen 11. Sistema de captación, procesamiento y visualización de datos.

Fuente: Autor

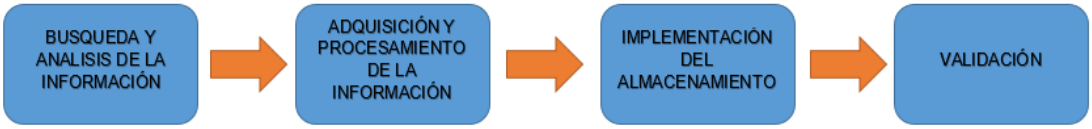
En la Imagen 12 se muestra el diagrama de los módulos en donde se puede observar los procesos necesarios para la realización del trabajo de grado. Iniciando con el módulo de Búsqueda y análisis de información. En esta etapa se realizará la búsqueda y análisis de todos los dispositivos capaces de la Adquisición de los niveles de glucosa, y otras variables que sean seleccionadas.

También se realiza la búsqueda de trabajos relacionados con la adquisición de datos obtenidos de variables fisiológicas para los cálculos necesarios para la predicción de insulina. Como resultado de esta etapa se seleccionarán las variables más relevantes para este trabajo, así como la selección de los dispositivos comerciales a utilizar.

La siguiente etapa es de Adquisición y Procesamiento de los datos obtenidos; en esta etapa se realiza la implementación y/o adaptación del dispositivo o dispositivos comerciales capaces de obtener información del conjunto de variables fisiológicas seleccionadas. Posteriormente, la información será procesada para luego ser almacenada. En la tercera etapa se definirá todos los aspectos necesarios para el almacenamiento de los datos tales como: el gestor de la base de datos, selección de servidor, etc.

Por último, se integra los anteriores módulos para poder validar el funcionamiento del sistema. Todas las selecciones tecnológicas a nivel electrónico y telecomunicaciones ser harán basadas en el análisis realizado en la primera etapa (búsqueda y análisis de información). La búsqueda de información se hará en fuentes confiables como bases de datos indexadas, páginas web académicas, libros, revistas científicas, google academic, etc.

Imagen 12. Composición de los módulos de la metodología.



Fuente: Autor

2.5.3.1 Búsqueda y análisis de la información

Para iniciar el desarrollo del trabajo de grado, se inicia buscando toda la información relacionada con los métodos y tecnologías utilizadas para la adquisición y monitoreo de las variables de frecuencia cardiaca, cantidad de pasos, presión arterial, calidad del sueño y el nivel de glucosa, que se utilizan en la actualidad, seleccionando solo las que se consideren necesarias. También se realizará la búsqueda de información relacionada para el procesamiento de la información, la comunicación (transmisión), bases de datos existente, su selección y todo lo relacionado para poder almacenar la información.

2.5.3.2 Adquisición y procesamiento de los datos.

En la segunda fase, se iniciará las pruebas con los dispositivos comerciales seleccionados para la adquisición de los datos obtenidos de las variables fisiológicas seleccionadas en un paciente diagnosticado de DM. Realizando la implementación/adaptación de los dispositivos, para posteriormente procesar los datos obtenidos, como por ejemplo microcontroladores, serie de microcontroladores STM32Cube IDE, Arduino, y Raspberry entre muchos otros.

Todas las selecciones se realizarán teniendo en cuenta los requerimientos del sistema funcionales y no funcionales.

2.5.3.3 Implementación del almacenamiento de los datos

En la tercera fase, se hará la implementación de la base de datos en el lenguaje de programación (por ejemplo: C, C++, C#, Java, etc.) (se revisó y recorto) y servidor seleccionado (como puede ser AWS entre muchos otros).

2.5.3.4 Validación del sistema

Para finalizar el proceso de desarrollo en esta fase se integrará el sistema de monitoreo, el cual está compuesto por los módulos de adquisición, procesamiento, almacenamiento de los datos. Para poder avanzar a la validación del sistema de monitoreo, donde cada etapa definida en la metodología será validada.

3. DISEÑO METODOLÓGICO

3.1 BÚSQUEDA Y ANÁLISIS DE LA INFORMACIÓN

En la etapa bloque de búsqueda y análisis, las soluciones propuestas en los antecedentes, se selecciona el dispositivo Freestyle libre de Abbott, el cual es un monitor continuo de glucosa (MCG), que no es invasivo lo que permite llevar un monitoreo de sus niveles de glucosa más ameno para el paciente, evitando punciones. La selección de este sensor está basada en su disponibilidad en Colombia y precio ya que ronda alrededor de 200.000 pesos colombianos. Estos sensores cuentan con una duración de 14 días por lo que se requiere de dos sensores para abarcar la duración del mes completo.

Solución propuesta para la selección del medidor continuo de glucosa, específicamente con el análisis de Pamela Apablaza justifica el uso del uso de sensores Freestyle para el monitoreo remoto de glucosa como se observa en la tabla 3.

Tabla 3. Tipos de monitor continuo de glucosa (MCG).

Tipo	Caracteristicas	Modelo (Marca)
MCG Profesional	- Lecturas de glicemia ciegas para el paciente - Utilizado como examen tipo "Holter de flicemia" - La información se obtiene en forma retrospectivas una vez que se retiró el MCG - Programa computacional entrega información como reportes que incluye gráficos, curvas y tablas	-ipro (Medtronics) - Freestyle libre (Abbott)
MCG-TR o personal sin equipo sin equipo infusor continuo de insulina	- De propiedad del paciente - Receptor del MCG con pantalla que muestra los niveles de glicemia antersticial y grafica su tendencia - Algunas marcas con transmision de la señal a smartphone, que actáa como receptor - Alarmas programables para alerta de niveles o variaciones de glicemia potencialmente riesgosas para el paciente - No esta integrado al ICSI - Se debe calibrar con niveles de glicemia capilar de 2 a 4 veces al dia	- G4 Platinium (Dexcom) - G5 Movile (Dexcom) - MiniMed Connect (Medtronics)
MCG-TR o personal sin equipo con equipo infusor continuo de insulina (SAPT)	- De propiedad del paciente - Equipo dual integrado con ICSI con pantalla que muestra los nivlees de glicemia intersticial y grafica su tendencia - Puede tener o no sistemas de suspensión automática de la infusión de insulina - Alarmas programables para alertar niveles o variaciones de glicemia potencialmente riesgosas para el paciente. - Se debe calibrar con niveles de glicemia capilar 2 a 4 veces al dia	- Paradim VEO (Medtronics) - MiniMed 640G (Medtronics) - Vipe (Animas) - T-slim G4 (Tadem)
MCG con sistema Flash	- Mide glicemias en tiempo real, pero no tiene receptor con pantalla que grafique todo el tiempo la informacion de la glicemia - La informacion es obtenida al pasar cerca del sensor el receptor, que funciona como lector tipo "scaner" - Sin alarmas programables (en el sensor), solo en las aplicaciones	- Freestyle libre (Abbott)

Fuente: Pamela Apablaza, Nestor Soto, Ethel Codner.

También se selecciona el reloj Fitbit tomando como referencia que este dispositivo es utilizado para la telemetría de los valores de frecuencia, cantidad de pasos y calorías de una persona. Este dispositivo previamente también fue evaluado por la universidad complutense de Madrid, y es utilizado actualmente por ellos para su investigación de algoritmos de predicción para el suministro de insulina. Por último, se decide utilizar una báscula con el objetivo de poder evaluar la evolución de la masa corporal del paciente y así el paciente y profesional analizar el respectivo proceso.

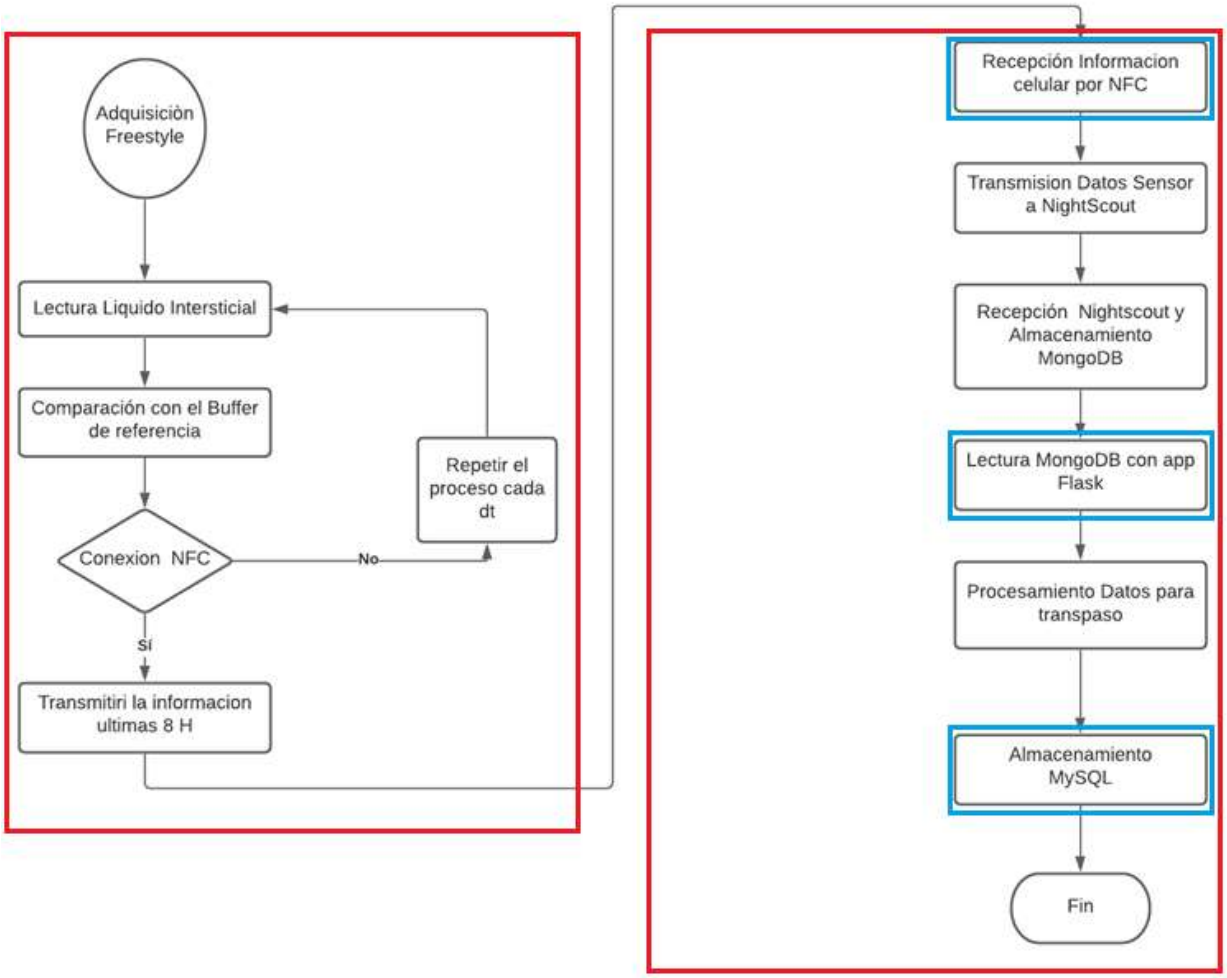
Una vez se hace la selección de los dispositivos

- Freestyle Libre (Nivel de glucosa en liquido intersticial).
- Fitbit (Lecturas de: Frecuencia cardiaca, cantidad de pasos y calorías).
- Bascula (Peso)

Se procede a realizar el diseño de la adquisición, procesamiento y almacenamiento de las variables fisiológicas que permiten monitorear la evolución del metabolismo del paciente diabético, para que así el profesional de la salud y el paciente puedan tomar medidas en su tratamiento.

Por lo anterior se procede a mostrar los diseños de la adquisición de los dispositivos, en el que se mostrara como se realiza la adquisición de los valores sensados por los dispositivos comerciales y su respectiva transmisión iniciando por la adquisición del nivel de glucosa a través del sensor de Freestyle que se observa en la imagen 13.

Imagen 13. Diseño Adquisición Glucosa intersticial.

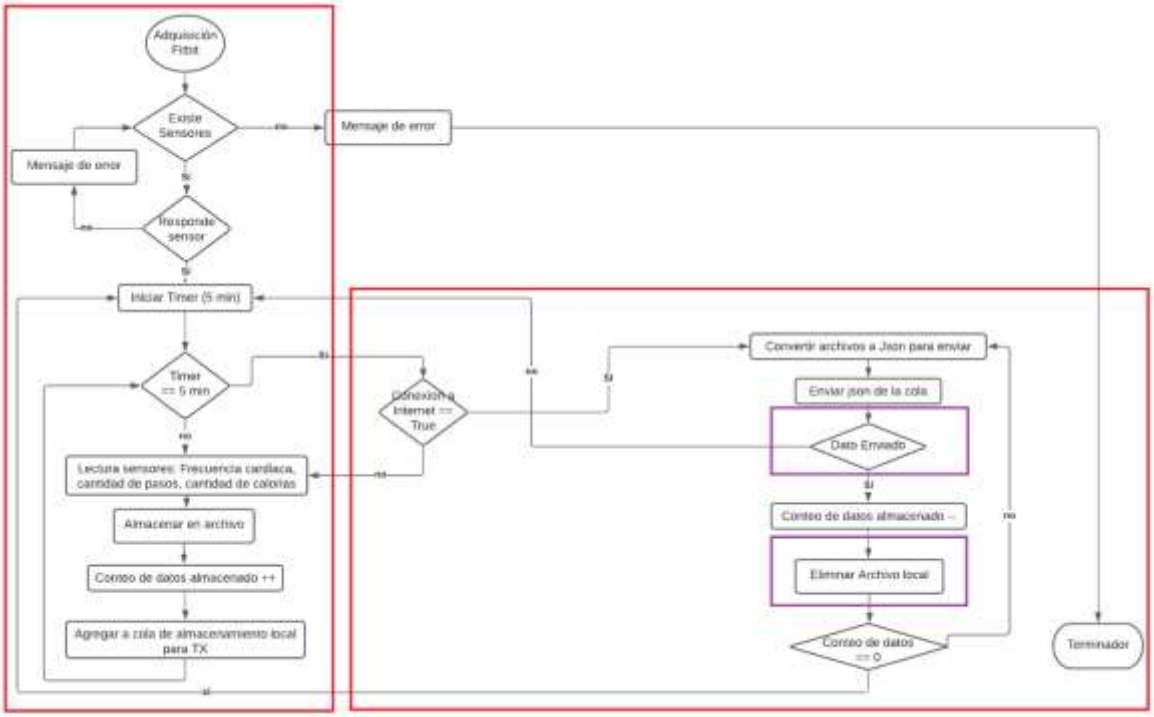


Fuente: Autor.

En la imagen 13, el diseño se divide en dos, en donde se muestra el proceso completo realizado por el sensor para la adquisición de los valores de glucosa y se transmite hacia el celular por medio de la tecnología NFC en el lado Izquierdo, mientras que en el lado derecho se muestra como es recibido los datos por medio de NFC (El dispositivo debe contar con esta tecnología), lo transmite a la aplicación que almacena los valores de glucosa en la base de datos MongoDB, para su posterior lectura y almacenamiento en la base MySQL donde se almacenaran los demás valores sensados.

Continuando con el diseño se procede a mostrar el diseño para el reloj Fitbit que al igual que el anterior, se divide en dos partes como se muestra en la Imagen 14.

Imagen 14. Diseño Adquisición conjunto de variables Fitbit.



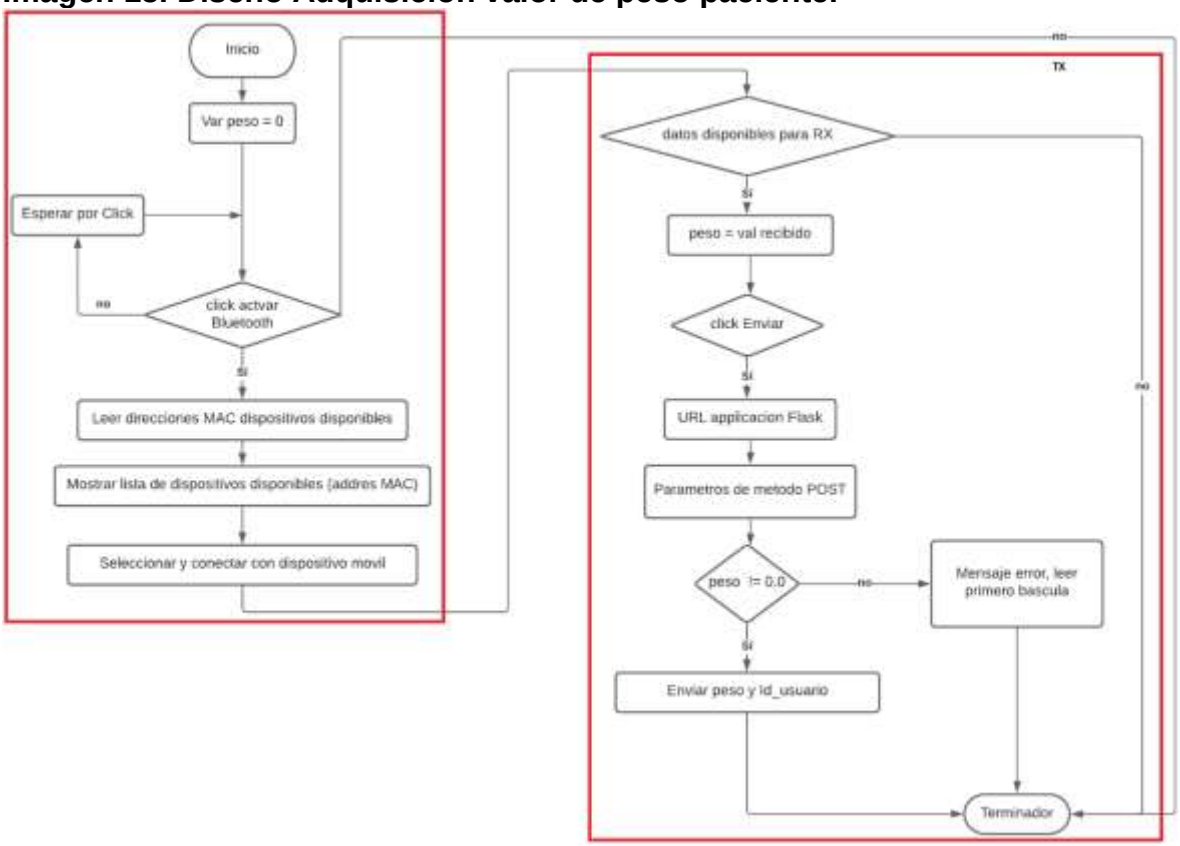
Fuente: Autor.

El diseño que se muestra en la Imagen 14, en la parte izquierda primero se realiza la validación de la existencia de los sensores en el dispositivo, una vez que se obtiene una respuesta de confirmación, se procede a evaluar si estos dispositivos están en funcionamiento y responden correctamente al dispositivo, en caso de que le confirme también al dispositivo, se procede al proceso del conteo (Timer) de 5 minutos para prueba de conexión y transmisión, en donde se evalúa si este tiempo ya se cumplió y en caso de que no sea así se almacena los valores de forma local para su posterior transmisión

En el caso contrario, cuando el tiempo de los 5 minutos se ha cumplido se valida si hay conexión a internet y si es confirmado esto, se procede a preparar los datos para la transmisión convirtiéndolos en formato Json.

La conexión es asíncrona, y es por esta razón que se realiza una validación de los datos enviados, una vez enviados los datos se hace el borrado local de los datos y se procede al siguiente dato a enviar. Los datos son enviados a la aplicación que se destinó para la recolección y visualización de todas las variables fisiológicas seleccionadas para este trabajo de grado.

Imagen 15. Diseño Adquisición valor de peso paciente.



Fuente: Autor.

Por último, en la imagen 15, se observar el diseño que permite la adquisición de los valores de peos de los pacientes diabéticos, dividido en dos partes. La primera parte en al costado izquierdo, es donde se realiza la sincronización con el dispositivo móvil. Mientras que en el costado izquierdo se realiza el proceso de validación de datos recibidos, en caso tal de tener valores de peos en el buffer de recepción, se realiza todo el procesamiento necesario para la transmisión de los datos del peso a la aplicación desplegada en la nube y que se encargara del almacenamiento de estos valores.

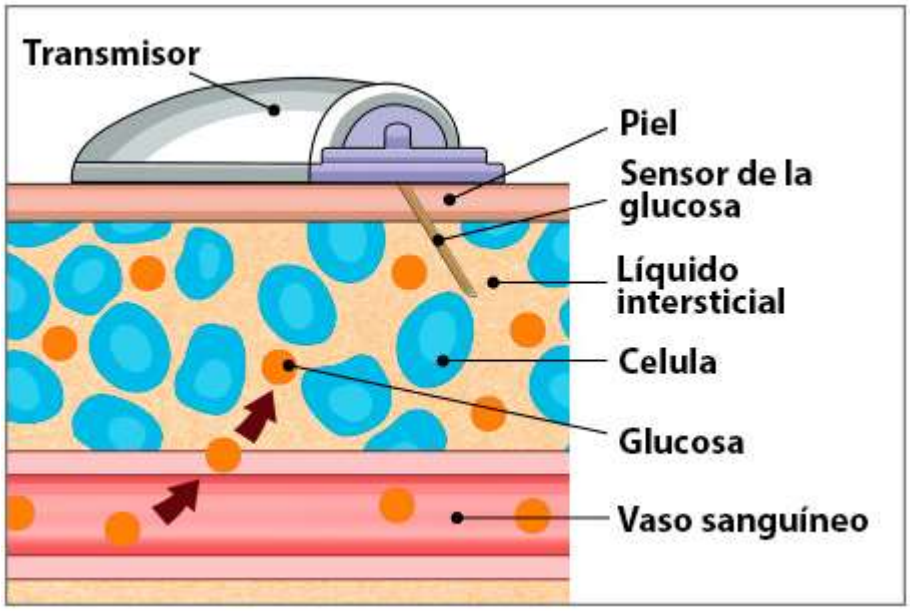
3.2 ADQUISICIÓN Y PROCESAMIENTO DE LOS DATOS.

3.2.1 ADQUISICIÓN.

3.2.1.1 Adquisición Freestyle libre

El sensor Freestyle es un sensor que mide el líquido intersticial para calcular la glucosa, como se observa en la Imagen 16. Es un sensor tipo parche lo que quiere decir que se aplica en la parte posterior del brazo con el aplicador especifico que brinda el fabricante, y el sensor se sujeta a la zona durante 14 días, una vez pasado esta cantidad de días, se debe retirar el sensor, por otro lado, hay realizar una medición de glucosa en el líquido intersticial, hay un retardo aproximado de 5 minutos respecto a la glucosa que se encuentra en la sangre. Una vez expresado lo anterior, se procede a la explicación del método de adquisición del sensor Freestyle Libre.

Imagen 16. Sensor y medición liquido intersticial.



Fuente: https://www.medtronicdiabeteslatino.com/recursos-y-ayuda/sensores-y-mcg/por-que-el-sensor-de-glucosa-no-siempre-coincide-con-la-glucosa.

FreeStyle logra una buena relación señal a ruido (en comparación con las tiras de reacción miniaturizadas) en volúmenes de muestra de solo 0,3 µL a través de una combinación de características de diseño distintas a las tradicionales al igual que sus métodos analíticos:

1. El Freestyle Libre, para aumentar la magnitud de la señal en su método de análisis utiliza la técnica electroquímica de coulometría. En contraste los sistemas de control de glucosa en la sangre que son tradicionales utilizan

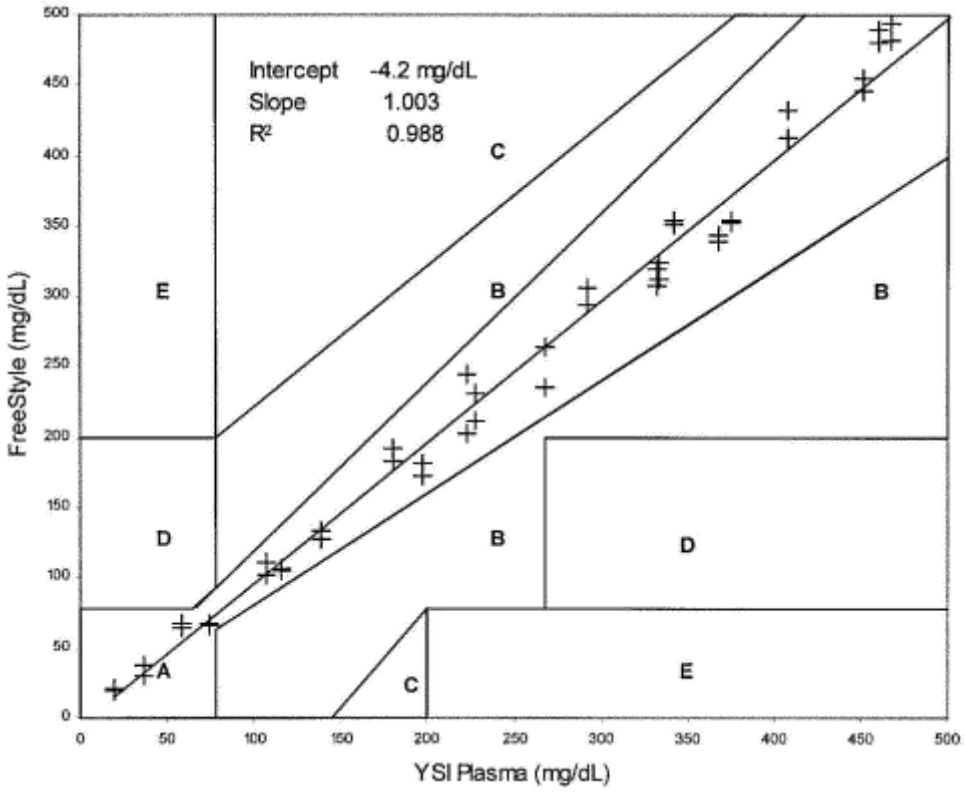
amperometría. Al utilizar un distinto método de análisis con respecto a la amperometría, se supera un inconveniente que presenta este método con las muestras de sangre ya que este método solo genera reacciones a una parte de la glucosa de la muestra, afectando a la corriente ya que esta se mide en un punto en un diferencial de tiempo dentro de esta reacción. Por el contrario, en la criometría se consigue se consigue hare reaccionar toda la glucosa de la muestra, midiendo la carga total generada por la reacción. Es por esta razón que resulta mucho más eficiente la señal obtenida por el método utilizado por el sensor Freestyle libre para mediciones de pequeñas muestras en comparación con el método tradicional con amperometría [64].

2. Los errores que se presentaban con el método tradicional, siendo afectados por la temperatura y el hematocrito, modificando la velocidad de reacción (en otras palabas la corriente), no afectando la carga total por lo que el método utilizado por el Freestyle Libre puede funcionar correctamente [65].

Ben Feldman, Geoff Mcgarraugh, Adam Heller, Nancy Bohannon, Jay Skyler, Eileen Deleeuw, y Dana Clarke, en su publicación, entregan una tabla en la que se muestra los resultados (ver Imagen 17) al realizar la medición de la carga generada por la reacción de toda la glucosa en la muestra, en donde el objetivo es reafirmar lo anteriormente mencionado, donde se resalta la ventaja del método de medición utilizado en diseño del sensor Freestyle Libre, mostrando que no tienen ninguna dependencia a el establecimiento de la corriente de estado estable.

[64] B. Feldman. "Freestyle(TM): A small-volume electrochemical glucose sensor for home blood glucose testing," Diabetes Technol. En: Rev. 2000. Ther. vol. 2, No. 2, pp. 222. doi: 10.1089/15209150050025177.
[65] Ibid., p. 222.

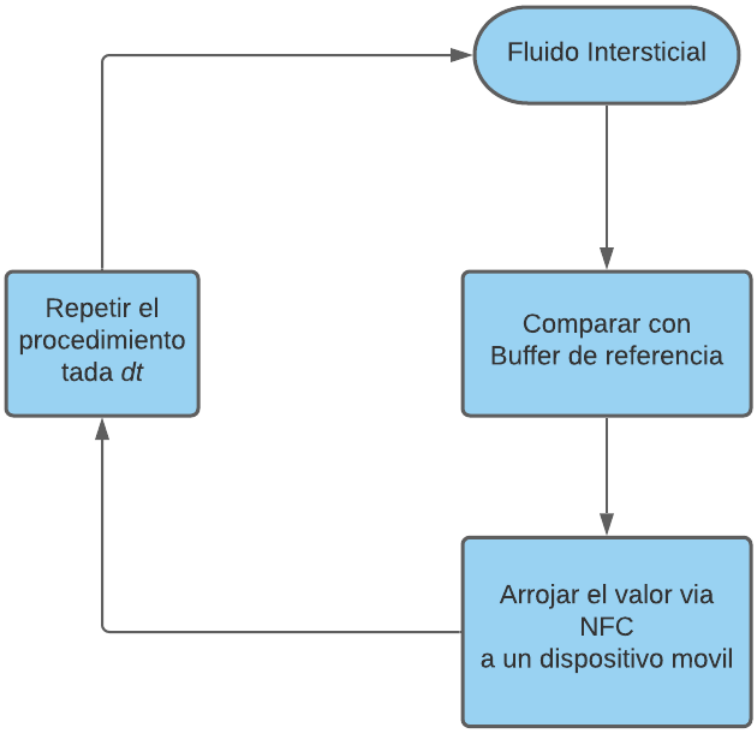
Imagen 17. Resultados análisis de carga FreeStyle.



Fuente: FreeStyleTM: A Small-Volume Electrochemical Glucose Sensor for Home Blood Glucose Testing.

En otras palabas, lo que quieren exponer es que la tecnología de Freestyle permite el control de una relación señal a ruido alta, aun cuando los volúmenes de las muestras se reducen en su magnitud. A continuación, se explica el funcionamiento interno de adquisición del Freestyle Libre por medio de la imagen 18.

Imagen 18. Esquema de funcionamiento interno Freestyle Libre.



Fuente: Reporte final BISU.

El esquema de funcionamiento de lectura y transmisión del Sensor Freestyle Libre comienza con el proceso de conversión de la glucosa en la muestra por medio de micrófitos a una carga que será leída por el sensor a través de las micro agujas (ver imagen 18).

El valor obtenido se compara con el buffer de referencia, para su posterior almacenamiento y transmisión al dispositivo móvil por medio de la tecnología NFC. Este proceso se repite cíclicamente en el periodo de los 14 días de uso del sensor antes de ser desechado.

El proceso de adquisición de los valores sensados por el sensor inicia con la aplicación en la parte posterior del brazo, para esto se debe realizar la validación recomendada por el fabricante del sensor Freestyle libre de comparar los códigos del sensor y del aplicador (como se observa en la imagen 19), deben ser los mismos de lo contrario no se debe aplicar.

Imagen 19. Freestyle.

Fuente: Autor.

Una vez que se corrobora que los códigos del aplicador y el sensor son los mismos se procede la aplicación del sensor en la parte posterior del brazo derecho a la persona de prueba (ver imagen 20) para realizar las lecturas de los niveles de glucosa registrados por el paciente en la aplicación Glimp en el caso de dispositivos Android.

Imagen 20. Sensor Freestyle aplicado.

Fuente: Autor.

Una vez que se leen los datos con la aplicación, al realizar el proceso de sincronización con la aplicación web Nightscout proporcionada por la comunidad de Nightscout se almacenarán estos valores en la base de datos MongoDB y posteriormente en MySQL, como se explica a continuación en el documento, permitiendo realizar un seguimiento remoto de los niveles de glucosa.

3.2.1.2 Adquisición Bascula.

El siguiente proceso de obtención de la variable física del peso es tomado de Fernandez Torres, Martin Abraham, Castro Sanchez, Rogelio en su trabajo “Prototipo de circuito para medición de peso con auto cero”[66].

La bascula es un elemento del día a día que permite obtener el valor del peso de un objeto, para este trabajo consta de una plataforma horizontal de vidrio que soporta el objeto /persona al que se desea pesar. Esta bascula consta de 4 elementos (Galga Extensiométrica) que convierten una deformación en una señal eléctrica[67].

Como se mencionó en el párrafo anterior, la ganga extensiométrica es la encargada de convertir la deformación en una señal. La galga extensiométrica es un transductor cuya resistencia cambia al aplicarse una deformación, esta característica es la que hace que sean elegidas para el sensado de peso, para este trabajo se utilizan 4 celdas de carga por lo que se conectan las cuatro a un puente Wheatstone, siendo cada una de las cuatro celdas de carga un componente del puente como se muestra en el capítulo de implementación de la báscula. Estas celdas de carga son alimentadas por un suministro de baja potencia de 5 V, entregada por la placa Arduino uno.

Las cuatro celdas de carga se conectan a un módulo amplificador HX711, el cual tiene la funcionalidad extra de conversor análogo digital de 24 bits, su diseñado está orientado para balanzas o basculas conectadas directamente a los sensores del puente. El módulo HX711 tiene una ganancia que es programable, también la velocidad de los datos 10 SPS (samples por second) y 8 SPS, y su alimentación se encuentra en el rango de 2.6 V hasta 5.5 V con una corriente de 1.5 mA y el rango de temperatura de funcionamiento correcto es entre -40°C y 85°C. Este módulo es conectado a la salida de las celdas de carga tal como se muestra en la implementación de la báscula, para poder realizar el procesamiento digital a través de la placa Arduino uno que se conecta a este módulo a través de dos pines para la sincronización y obtención de los datos digitales enviados por el módulo.

3.2.1.3 Adquisición Fitbit

El Fitbit cuenta con un conjunto de sensores que están disponibles a través de un conjunto de librerías que el fabricante de Fitbit Ionic, Para utilizarlas solo hace falta exportar en el entorno de desarrollo. Hecho lo anterior se verifica que los sensores a los que se desea acceder a los datos están presentes en el dispositivo y responde.

---

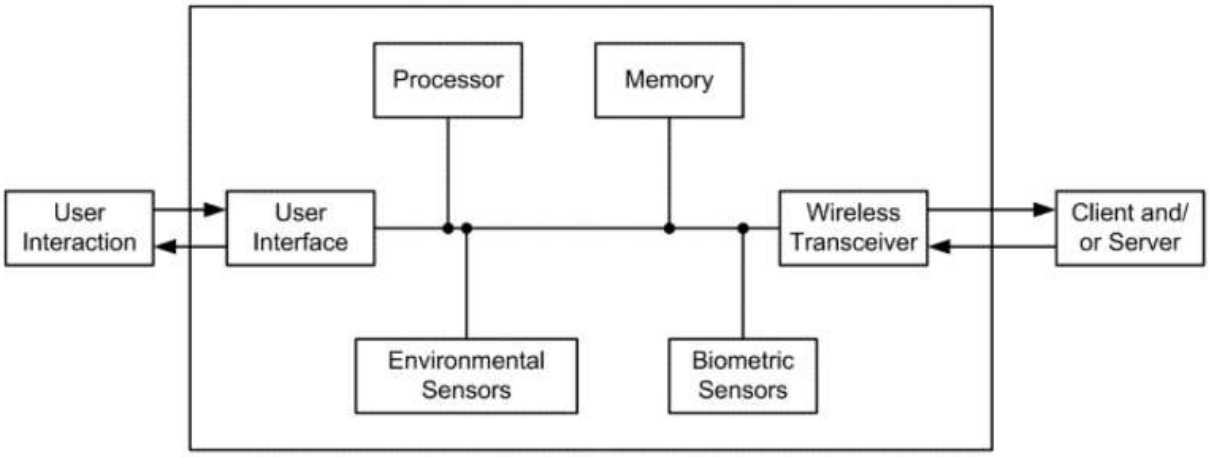
[66] C. IRAPUATO SALAMANCA. “Prototipo de circuito para medición de peso con auto cero”. En: Revista Jóvenes en la ciencia., No. 4, (2008); p. 3140.

[67] Ibid., p. 3140

Este proceso es exactamente igual para las variables que se extraen para este trabajo de grado (frecuencia cardiaca, cantidad de pasos y cantidad de calorías).

El esquema que se muestra a continuación en la imagen 21. Muestra el esquema general de la sensorica integrada en el reglo Fitbit Ionic.

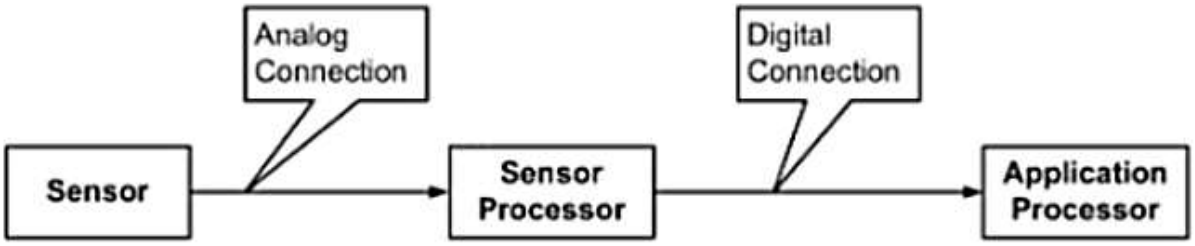
Imagen 21. Esquema general de Funcionamiento del reloj Fitbit.



Fuente: https://patents.google.com/patent/US10856744B2/en?q=fitbit&oq=fitbit+.

En cuanto a la secuencia interna que utiliza la arquitectura de los sensores se encuentra en la Imagen 22, donde inicia con el sensor, por medio de una conexión física separa el valor analógico al procesador dedicado a la sensorica del reloj encargado de recibir y procesar la información para posteriormente realizar su transmisión de forma digital

Imagen 22. Diagrama de secuencia proceso de extracción valor de los sensores.



Fuente: https://patents.google.com/patent/US10856744B2/en?q=fitbit&oq=fitbit+

3.2.2 PROCESAMIENTO.

En esta etapa, ingresan los datos tomados en la etapa anterior (etapa de adquisición), para realizar el procesamiento de los datos, para ser transmitidos y recibidos por una aplicación web desarrollada por este trabajo de grado, esta aplicación se encarga de recibir los datos enviados por los tres dispositivos (aclarar que del Freestyle lo hace a través de la aplicación web de Nightscout), para ser almacenado en una sola base de datos.

El procesamiento de los datos adquiridos por cada uno de los dispositivos es el siguiente:

3.2.2.1 Procesamiento de la báscula:

Una vez se obtiene el valor entregado por los sensores de báscula en milivoltios a el módulo HX711 (de las cuatro celdas de carga en una sola salida), este se encarga de la conversión de las variaciones de la señal entregada en las celdas de carga en valores digitales de 24 bits, por lo que el siguiente paso es la calibración de la escala, el cual consiste en la relación de la salida promedio del ADC dividido por el peso. Aprovechando el comportamiento lineal se obtiene el valor de factor de calibración lo que permite obtener el resultado en gramos. Para este trabajo de grado este proceso es realizado por la librería Bogdan Necula y Andrea Moti en Arduino que permite realizar el proceso anteriormente mencionado.

Una vez se obtiene el valor de peso retornado por la función propia de la librería mencionada anteriormente, este valor es preparado para ser enviado a través del módulo Bluetooth HC-06 hacia el dispositivo móvil sincronizado para realizar el proceso de envió y almacenamiento de la variable de peso.

3.2.2.2 Procesamiento del Freestyle:

El procesamiento que corresponde al sensor Freestyle, corresponde a la extracción de los valores leídos del sensor por el dispositivo móvil con la aplicación Glimp, esta aplicación se encargara de realizar la carga o subida de la información a la aplicación brindada por la comunidad de Nightscout orientada a personas diabéticas. Una vez los datos están alojados en la base de datos de la aplicación web de Nightscout, se procede a la realización al proceso de extracción de todos los valores almacenados allí, leyéndolos la aplicación web desarrollada en este trabajo de grado, una vez se obtiene los valores almacenados en otra base de datos, se preparan para ser enviados en cola a la base de datos MySql y son enviados.

3.2.2.3 Procesamiento del Fitbit:

Para el procesamiento del Fitbit, los datos obtenidos en la etapa de adquisición, estos serán almacenados en archivos Json, y a su vez se ira generando una cola y un conteo de archivos dentro del dispositivo para saber con exactitud cuántos son los archivos que se deben enviar al momento de tener conexión y realizar el envío de los mismos, la recopilación se realiza en la venta una ventana de 5 minutos, una vez acaban los 5 minutos (este conteo se realiza con el uso de timer dentro del dispositivo), se realiza una prueba de conexión para proceder con la transmisión a la aplicación web que se encargara de almacenar los datos en la base de datos, cada vez que se realiza el envío de un archivo Json que contiene los datos sensados por el Fitbit, se disminuye el contador de archivos interno y se elimina el archivo local enviado para evitar redundancia en los datos almacenados en la base de datos, y se continua con el siguiente archivo en la cola de espera. En caso de que no haya conexión lo que se hace es continuar con el conteo de los archivos que se están almacenando internamente en el dispositivo y volver a intentarlo dentro de los 5 minutos siguiente.

La conexión con la aplicación se realiza por medio de una dirección URL (Uniform Resource Identifier) generada por el servidor Heroku que ofrece hosting gratuito, además de una URL especifica especificada por la aplicación web diseñada en Flask que será “URL_HENERADA POR HEROKU/fitbit”, esta URL permite a la aplicación recibir los datos enviados por el reloj Fitbit Ionic, desfragmentar la información y prepararla para el almacenamiento en la base de datos.

3.3 IMPLEMENTACIÓN DEL ALMACENAMIENTO DE LOS DATOS

Para la implementación del almacenamiento de los datos, se realizó a través del servidor Heroku el cual ofrece servicios de conexión con servidores que prestan el servicio de almacenamiento de distintas bases de datos como PostgreSQL, MySQL, MongoDB, entre otros, para este trabajo de grados se seleccionó un servidor MySQL brindado por el servicio ClearDB dentro de Heroku, el cual brinda por defecto un usuario y contraseña únicos para el acceso a la base de datos, estos son necesarios para que en la etapa de procesamiento se envíen los datos al servidor y sean almacenados a través de consultas MySQL ya que la base de datos solo entiende estas instrucciones. Con lo anterior se garantiza el almacenamiento de los datos dentro de la base de datos.

3.4 VALIDACIÓN DEL SISTEMA

Para la validación, se realizan mediciones de las variables físicas con los tres dispositivos, una vez se obtiene los valores, se realiza la validación de su almacenamiento correcto, comparando el valor sensado con el valor almacenado en la base de datos (nivel de glucosa, cantidad de calorías (Calories Rate), cantidad de pasos (Steps Rate), frecuencia cardiaca (Heart Rate) y peso).

4. IMPACTO Y RESULTADOS ESPERADOS

Este trabajo de grado se espera que tenga un impacto multinivel abarcando el nivel económico, social y tecnológico. A continuación, se explica en detalle cada uno de ellos.

4.1 IMPACTO ECONÓMICO.

Se espera reducir costos al sector salud, al aumentar el número de pacientes que controlan su nivel de glucosa puesto que esto previene y reduce significativamente la probabilidad de desarrollar enfermedades crónicas o deterioro en el cuerpo como: puede dañar sus ojos, riñones, nervios, piel, corazón, y vasos sanguíneos, etc. Como resultado, el control del nivel de glucosa y el conjunto de variables fisiológicas expresadas en este trabajo de grado, reduce los costos de tratamiento a los pacientes, evitando su desplazamiento a centro médicos para el seguimiento por parte del profesional de la salud en el tratamiento.

4.1.1 IMPACTO SOCIAL.

Se espera que el uso del sistema tecnológico propuesto en este trabajo de grado mejore la calidad de vida de las personas diabéticas al hacer un seguimiento confiable de nivel de insulina, así como también las otras variables incluidas en este trabajo de grado, permitiendo que el tratamiento médico de los pacientes diabéticos puede ser ajustado acorde su condición en un menor tiempo al tener el profesional un acceso más rápido y continuo a la información del paciente.

Por otra parte, también se espera mejorar la calidad de la pronta atención en casos de urgencia, como bien se sabe la diabetes es una enfermedad que con el paso del tiempo afecta diferentes órganos del cuerpo si no es tratada adecuadamente (ejemplo pie diabético).

Este trabajo de grado tiene limitaciones geográficas dentro del territorio colombiano siempre y cuando se pueda tener acceso a Internet, lo que permite que persona que residen en zonas rurales puedan tener un mejor monitoreo y calidad de vida evitando desplazamientos a centros médicos para realizar el control de la enfermedad, este punto de vista se hace aún más importante en las persona adultas mayores que padecen de esta enfermedad debido a que su movilidad por la edad y las características de la propia enfermedad les dificulta grandes desplazamientos y a su vez los lleva a un estado de dependencia de sus familiares aún mayor.

4.1.2 IMPACTO TECNOLÓGICO.

Con el conjunto de dispositivos tecnológicos comerciales que integran el sistema de adquisición E-Health para personas diabéticas, se adquieren mediciones confiables y proporciona un conjunto de información completo sobre la información de nivel de glucosa y otras variables importantes para el control de los pacientes diabéticos, que va desde la toma de la medida del conjunto de variables hasta una acción por parte del servicio médico, en caso de ser requerido.

Este es un sistema desarrollado a nivel académico da una base para posteriores investigaciones en temas no solo tecnológicos sino también médicos integrando estas y muchas más disciplinas que permitan llevar a cabo un mejor monitoreo remoto de los pacientes diabéticos. Con base en esta primera entrega se seguirán realizando mejoras hasta alcanzar un productivo final con un desarrollo tecnológico realizado por estudiantes de la Universidad Católica de Colombia.

5. DESCRIPCIÓN LOS COMPONENTES

5.1 COMPONENTES DE HARDWARE

5.1.1 ARDUINO UNO

Para la selección del microprocesador se comparan las distintas tarjetas de desarrollo que se pueden usar de acuerdo a la tabla 4.

Tabla 4. Tabla comparativa entre distintas tarjetas de desarrollo.

Microcontrolador	pines	Memoria (KB)	Programación
Arduino Mega	54	256	C++
Arduino Uno	14	33	C++
Raspberry Pi versión 1	26 GPIO	256000	Python
STM32f446 núcleo 64	64-144	256-512	C++

Fuente: Autor.

En la tabla 4, se hace la comparación entre 4 microcontroladores y se escoge el Arduino uno, ya que las características técnicas que posee se ajustan a las necesidades del trabajo de grado. Tales como la cantidad de pines suficientes (en el caso del Arduino uno posee 14) y cantidad de memoria disponible. Por otro parte, el lenguaje de programación con el que se trabaja la tarjeta Arduino Uno es C++. Este leguaje es conocido por el autor para el desarrollo en microcontroladores y a su vez al ser un lenguaje que es compilado directamente, y no interpretado como Python ofrece un mayor rendimiento al momento de desarrollar y utilizar los recursos directamente del Hardware.

Estas características son suficientes para el proceso de adquisición y procesamiento de la información desde la placa Arduino Uno. El microcontrolador que utiliza esta tarjeta de desarrollo es el microcontrolador ATmega328p, contando con 6 entradas análogas, 14 pines de entrada y salida digital, de los cuales 6 pueden ser usados para aplicaciones de PWM, un cristal de 16MHz utilizado para generar la señal de reloj necesaria para el funcionamiento del procesador, conexión USB (0 y 5 V), conector Jack de alimentación (0 y 12 V). Cualquiera de los dos conectores se puede utilizar para alimentar la placa, con la diferencia de que el puerto USB tiene protección contra picos y voltajes invertidos, mientras que la conexión con el Jack no cuenta con ningún mecanismo de protección y es necesario alimentarlo con una fuente que este correctamente regulada [68].

[68] 53ARDUINO UNO. descripción de dispositivos de arduino {En línea}. {10 noviembre de 2020} disponible en: http://arduino.cl/arduino-uno/

5.1.2 SENSOR FREESTYLE LIBRE ABBOTT.

Para la selección del sensor se tomó en cuenta los disponible en el mercado comparándolos en la tabla 5.

Tabla 5. Sensores de glucosa en el mercado

Nombre sensor	Días de funcionamiento	Accesibilidad en Colombia
Freestyle Libre	14	5
Medtronic	14	2
Dexcom G5/G6	14	3

Fuente: Autor.

En la tabla 5, el campo de accesibilidad hace referencia a que tan fácil es acceder a uno de estos dispositivos, donde va la escala de 1 a 5, lo que quiere decir que el sensor con mayor accesibilidad en Colombia porque se pueden adquirir directamente en el país y su precio es inferior al de la competencia. Por la anterior razón y las características que posee el sensor, se seleccionó los sensores Freestyle Libre para el desarrollo de este trabajo de grado.

El sensor Freestyle Libre cuenta con unas dimensiones pequeñas (35 mm x 35 mm), su duración es de 14 días, pasado este periodo se debe cambiar de sensor, es calibrado desde fabrica, por lo que no requiere de pinchazo en el dedo para ello, mecanismo de adherencia la piel utilizada por este sensor permite llevarlo mientras se baña, nada o hace ejercicio [69].

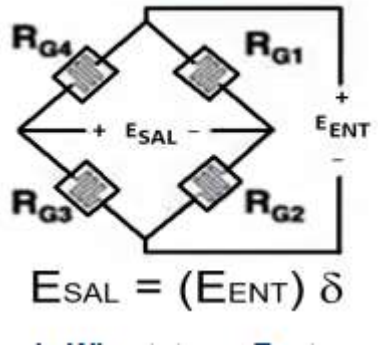
5.1.3 CELDAS DE CARGA.

Una galga extensiométrica es un transductor, con la particularidad de que su resistencia varia al suministrar esfuerzo o deformación en una celda de carga, y estas celdas de carga son las que se conectan al puente o Wheatstone [70]. Para este laboratorio se van a colocar cuatro resistencias en el puente, así como se muestra en la imagen 23.

[69] Comprar FreeStyle Libre Sensor Lector En Farmalisto Colombia. {En línea} {10 noviembre 2020] disponible en: https://www.farmalisto.com.co/freestyle/39473-comprar-freestyle-libre-sensor-lector-de-glucosa-caja-con-1-unidad-precio-precio-5021791000432.html?gclid=CjwKCAiAtK79BRAIEiwA4OskBmlYJPxJ6TbL81H7FMbFSYkP7f79e2IFV8o17ayQfZMR1GgRibjEMRoCD5kQAvD_BwE

[70] C. IRAPUATO SALAMANCA. Óp. cit., p. 3141

Imagen 23. Puente Wheatstone.



Fuente: Autor

5.1.4 UN MÓDULO HX711 AMPLIFICADOR (ADC) PARA LAS CELDAS DE CARGA.

El módulo HX711 es seleccionado para este trabajo debido a que está diseñado para interactuar directamente con el sensor funcionando como amplificador y ADC para balanzas de peso dentro de la industria. El módulo HX711 Genera salidas digitales de 24 bits, también cuenta con características como una ganancia programable para las dos secciones de velocidades 10 SPS (Simples Per Second) y 8 SPS. Su alimentación debe estar en el rango de 2.6 V hasta 5.5 V/1.5 mA, y su rango de temperatura de -40°C y 85°C

5.1.5 MÓDULO HC-06 BLUETOOTH.

El módulo seleccionado para la transmisión de la información de la báscula hacia el dispositivo móvil es el HC-06, el cual es un dispositivo transceptor inalámbrico con un BER aproximado de -80 dBm, con una potencia de radiación en la salida entre el rango de -dBm a +60 dBm[71]. Las especificaciones de la cuarta versión de Bluetooth, define características interesantes para usos IoT, así como una interfaz de radio frecuencia (RF) de comunicaciones inalámbricas y el conjunto de protocolos contemplado en la norma IEEE802.3.

[71] J. JUNG. “HC-06 Bluetooth based driver module for emergency LED Multi-Directional Indicator” Vol 12 No. 10052197, (2017); p. 115.

5.1.6 DISPOSITIVO MÓVIL CON SISTEMA OPERATIVO ANDROID.

Para este trabajo de grado se seleccionó dispositivos móviles son sistema operativo Android debido a la gran cantidad de usuarios que poseen este sistema operativo, aumentado el número de personas que se puedan encontrar beneficiadas con el proceso realizado en este trabajo de grado.

5.1.7 RELOJ FITBIT.

El reloj Fitbit tiene una dimensión de 29.232 x 21 mm, diagonalmente sus dimensiones son 35.99, grosor de 12mm y un peso de 46 g. cuenta con los siguientes sensores y componentes: acelerómetro de tres ejes, giroscopio de tres ejes, monitor óptico de ritmo cardiaco, altímetro, sensor de luz ambiental, motor de vibración. También cuenta con una memoria de 2.5 GB, modulo wifi (802.11/b/g/n) [72].

Todas las características del Fitbit, así como su sensoria integrada es la que permite realizar la extracción de datos de las variables fisiológicas de frecuencia cardiaca, cantidad de pasos y las calorías, necesarias para el monitoreo y control de pacientes diabéticos.

5.2 COMPONENTES DE SOFTWARE

5.2.1 SERVIDOR PARA ALMACENAR Y EJECUTAR LA APLICACIÓN WEB (AZURE, HEROKU, AWS).

Es una plataforma en la nube que permite a las empresas o proyectos construir, entregar y alojarlas en la nube. La selección de este servicio en la nube está relacionada con el hecho de que este permite desplegar aplicaciones conectando con Git para poder tener un registro de versiones, y por otro lado es su plan gratuito, al ser gratuito tiene sus restricciones, pero las aplicaciones desplegadas dentro de él se ejecutan bien.

[72]Fitbit Ionic, análisis: Fitbit acierta con su segundo smarkwatch, que además deslumbra en autonomia {En línea} {10 noviembre de 2020} disponible en: https://www.xataka.com/analisis/fitbit-ionic-analisis-caracteristicas-precio-especificaciones

5.2.2 CONTRIBUCIÓN DE LA COMUNIDAD NIGHTSCOUT (APLICACIÓN WEB).

La comunidad de Nightscout, desarrollan una aplicación que dejan abierta al público para el seguimiento remoto de insulina, esta aplicación se debe desplegar en un servidor, pero la comunidad recomendó Heroku.

5.2.3 CREAR CUENTA DE LOS SIGUIENTES SERVICIOS ON CLUD:

Para llevar a cabo todo el proceso es necesario crear una cuenta de los siguientes servicios.

- Heroku, App Inventor, GitHub, MongoDB Atlas.
- Tecnología de desarrollo solución Flask.
- servidores de bases de datos MongoDB y MySQL.

6. IMPLEMENTACIÓN

En esta sección se brindará una explicación detallada de la implementación de cada uno de los módulos en los que se definió el desarrollo necesario para este trabajo de grado: Consisten en captación, procesamiento, transmisión y almacenamiento de las variables fisiológicas seleccionadas. En este caso, esta sección explica el funcionamiento completo del sistema como se observa en la Imagen 24, así como la implementación para obtener los datos de las variables fisiológicas del paciente, como se realiza el procesamiento de datos, su procesamiento y envío a un servidor remoto para realizar el proceso de almacenamiento.

Imagen 24. sistema E-Health.

Fuente: Autor.

6.1 IMPLEMENTACIÓN DE APP WEB NIGHTSCOUT EN SERVIDOR HEROKU.

En principio Nightscout, fue desarrollado por padres de niños con diabetes tipo 1, que actualmente sigue siendo desarrollada, con mantenimiento y apoyado por voluntarios. Nightscout es un proyecto de código abierto que permite el acceso en tiempo real a los datos de un CGM (Medidor Continuo de Glucosa, en español) a través de un sitio web personal, aplicaciones, visores de relojes inteligentes y widgets disponibles para teléfonos inteligentes.

El objetivo de este trabajo de grado es permitir a los usuarios (pacientes, como familiares y expertos de la salud) de los distintos sensores de monitorización de glucosa, su monitoreo remoto. Hoy en día Nightscout tiene conectividad y operatividad con los siguientes dispositivos de monitorización de glucosa: Dexcom G4, Dexcom share con el sistema operativo Android, Dexcom share/ G5 que trabaja con el sistema operativo iOS y al igual que Medtronic, con la diferencia de que este último también funciona con el sistema operativo Android y por último Freestyle libre. Este último es seleccionado para este trabajo de grado gracias a su precio y precisión, es un dispositivo más económico y accesible en Colombia teniendo distribuidores a nivel nacional, mientras que con los demás medidores de glucosa no se cuenta con este servicio y se deben pedir directamente a la empresa correspondiente.

- **¿Cómo se hace posible obtener el Monitoreo Continuo de Glucosa en la nube?**

Lo primero es identificar los siguientes aspectos para poder abordar la configuración correcta de Nightscout.

Qué tipo de teléfono portara el paciente con DM, ya que es necesario que el paciente tenga constantemente su dispositivo para poder realizar las mediciones de su nivel de glucosa. Aquí hay varios aspectos a analizar:

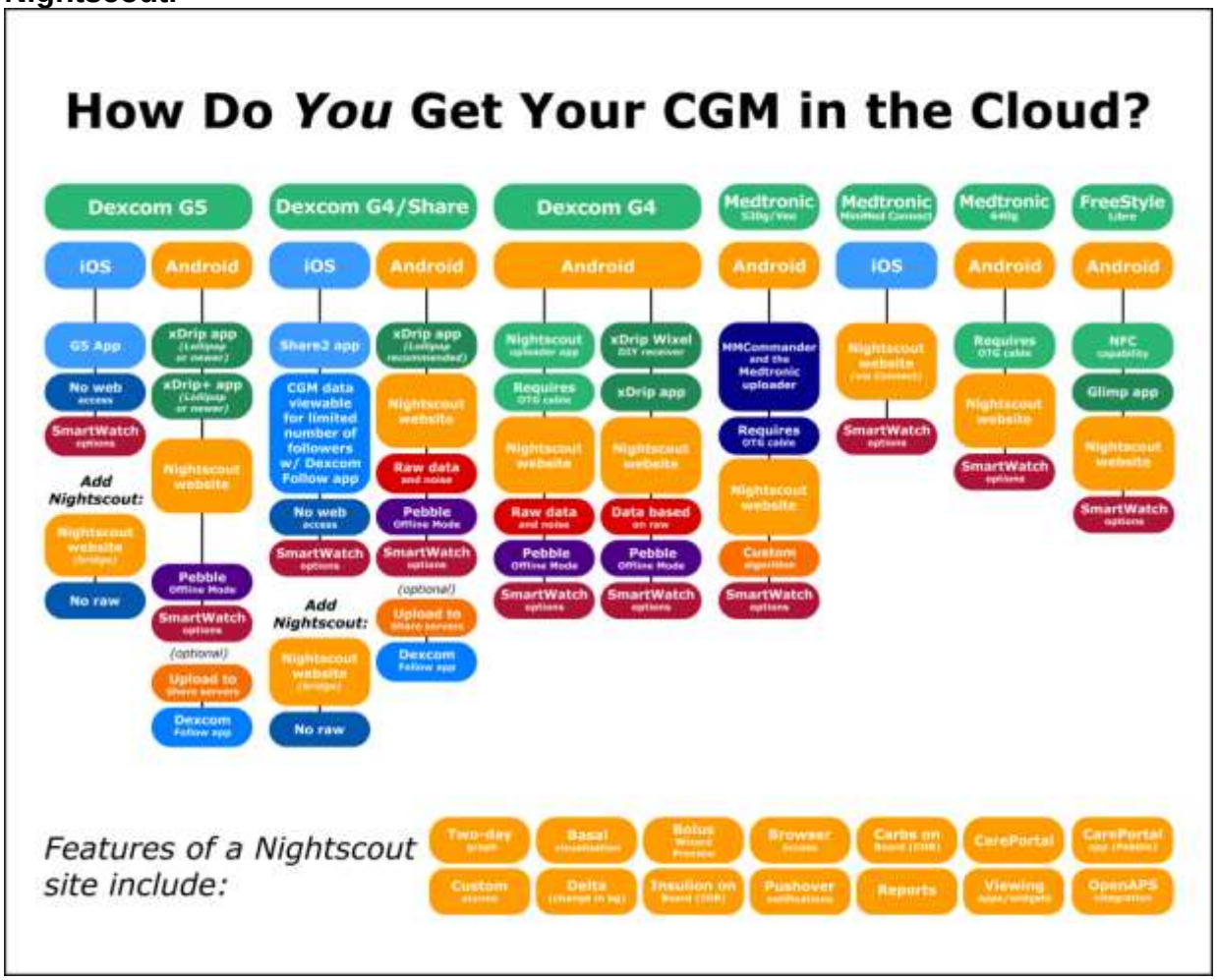
Si el dispositivo cuenta con NFC o no, para el caso de este trabajo de grado se hizo uso de un Xiaomi MI 9 T Pro con sistema operativo (OS) Android, el cual cuenta con esta tecnología. En este mismo punto se analiza el sistema operativo del dispositivo, ya que esto afecta en el proceso de extracción o lectura de los datos del sensor. Ya que hay una relación entre el sistema operativo y la implementación Nightscout en el servidor de Heroku.

- **¿Qué tipo de MCG se tiene o se desea utilizar?**

Para este trabajo de grado se seleccionó el Sensor Freestyle Libre versión 1, de la empresa de Abbott.

En la imagen 25 que se muestra a continuación la página de Nightscout brinda una descripción de las rutas disponibles para la configuración del monitor remoto del sensor de glucosa usando Nightscout[73].

Imagen 25. Rutas disponibles para la configuración del monitor remoto Nightscout.



Fuente: Comunidad de Nightscout página Principal: http://www.nightscout.info.

Para usar Nightscout con Freestyle libre es necesario lo siguiente:

- Un sensor FreeStyle libre en funcionamiento.
- Un sitio de Nightscout en funcionamiento.
- Un dispositivo con tecnología NFC con sistema operativo Android o iOS.
- La aplicación correspondiente al sistema operativo:

[73] "The Nightscout Project – We Are Not Waiting." {11 noviembre de 2020} disponible en: http://www.nightscout.info/.

- Glimp para Android.
- Spike para iOS.

- **Configuración de un sitio Nightscout**

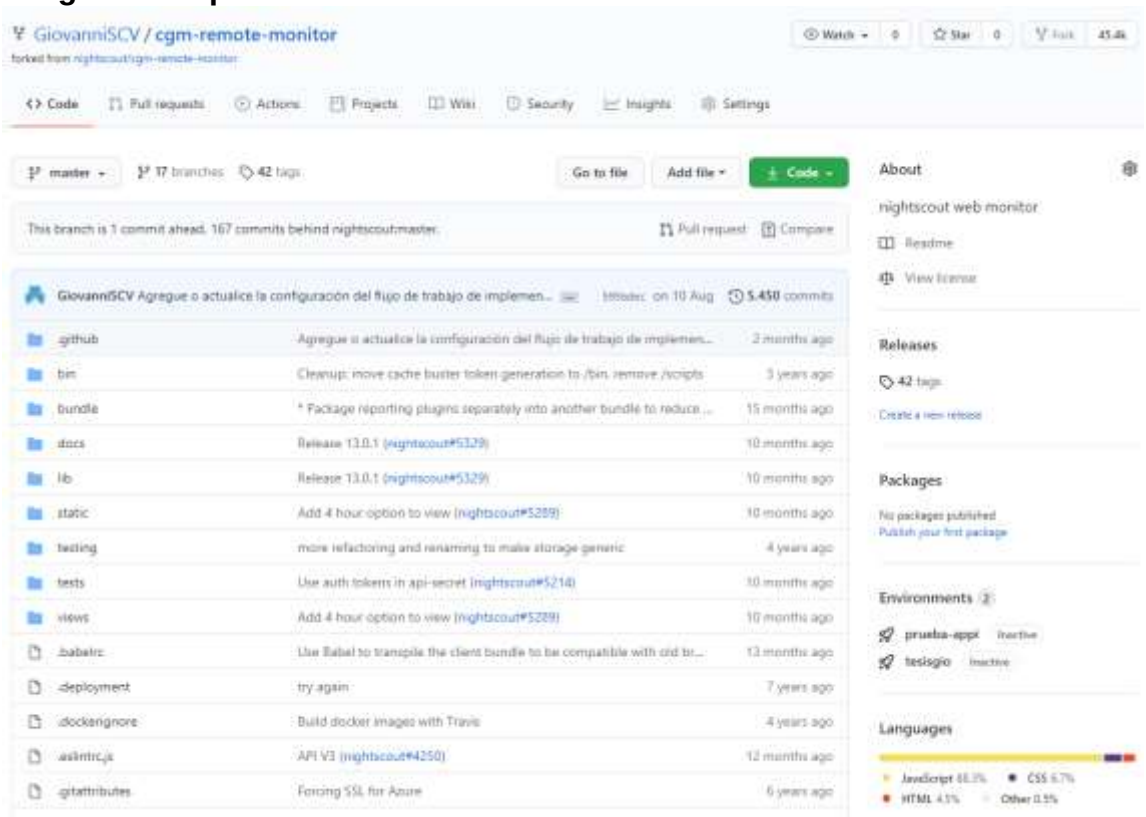
Dentro de la implementación del sitio Nightscout se debe tener en cuenta lo siguiente:

- CarePortal debe estar habilitado en Heroku o Azure.
- Debe tener un API_SECRET configurado en los servidores Heroku o Azure.

- **EXPLICAR LAS DOS VARIABLES**

Lo primero que se debe hacer es registrarse en la plataforma de GitHub y luego visitar el repositorio oficial de Nightscout (ver imagen 26) el cual se encuentra en https://github.com/nightscout/cgm-remote-monitor, una vez se encuentra se debe copiar este repositorio en nuestro perfil para poder realizar el Deploy, así como se muestra a continuación.

Imagen 26.Repositorio GitHub.



Fuente: https://github.com/nightscout/cgm-remote-monitor.

Al clonarlo (hacer una copia del proyecto[repositorio] en nuestro GitHub), en la parte de abajo existe un botón con los cuales procederemos a realizar el Deploy de la aplicación Web Nightscout como se muestra a continuación en la Imagen 27.

Imagen 27. Opciones de Deploy con tres servidores (Azure, Heroku y propio).

Fuente: nightscout/cgm-remote-monitor: nightscout web monitor (github.com).

Para este trabajo de grado se selecciona el “Deploy” en Heroku después de haber también realizado el Deploy en Azure, pero por problemas de permisos de acceso

de la aplicación Glimp con respecto a Azure, no fue posible continuar con este servidor por lo que se decidió retomar el servidor de Heroku, aunque este genere un error en la aplicación Glimp al momento de realizar la prueba de conexión, pero ya realizando la conexión Glimp con App Web Nightscout funciona correctamente.

Para poder continuar, se crea una cuenta en Heroku en donde es necesario agregar los datos de una tarjeta de crédito para poder seleccionar un plan para el servicio. Las opciones de planes disponibles son: Profesional (Estándar 1x/2x, rendimiento - M/L costo 25$ hasta 500$ dólares mes), pasatiempo ($7 mes), y gratuito **($0 mes)**. Para este desarrollo se selecciona el plan gratuito dado que este plan ofrece funciones básicas necesarias (ver Imagen 28).

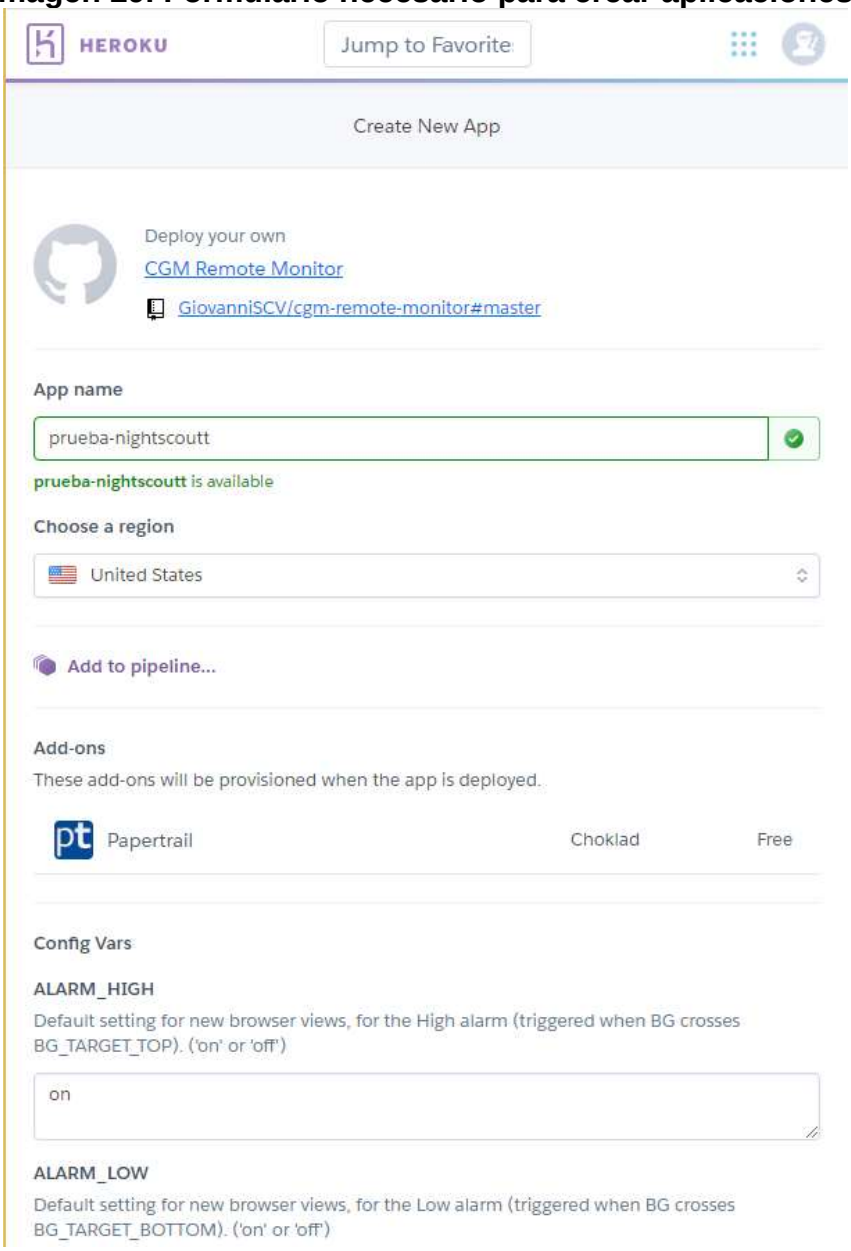
Imagen 28. Portafolio de planes disponibles Heroku.

Fuente: Autor desde dashboard Heroku.

Una vez creada la cuenta, el siguiente paso es especificar los valores de las variables del formulario que genera el servidor Heroku, seleccionando el botón “Deploy to Heroku”, como se muestra en la Imagen 27.

En el formulario es importante definir el nombre de la aplicación *prueba-nightscout* y la región que se selecciona es EE. UU. como se observa en la imagen 29.

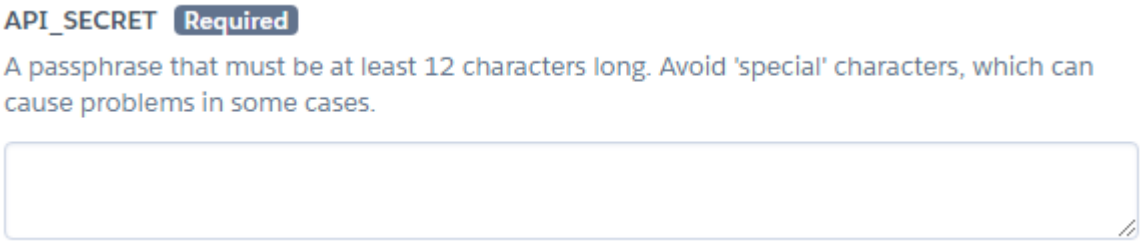
Imagen 29. Formulario necesario para crear aplicaciones.



Fuente: Autor desde dashboard Heroku.

En este formulario principalmente se debe llenar los campos de:

Imagen 30. Campo de API_SECRET.



Fuente: Autor desde dashboard Heroku.

El campo de API_SECRET como se observa en la imagen 30 es muy importante porque esta es la contraseña de nuestra aplicación. Hay que recordarla ya que esta después se utiliza para poder generar una conexión con la aplicación que se instala en el celular para poder realizar las lecturas del Sensor FreeStyle Libre Abbott, otro campo importante es el Campo ENABLE.

Imagen 31. Campo de ENABLE.

Fuente: Autor desde dashboard Heroku.

Este campo que se observa en la imagen 31, tiene la funcionalidad de habilitar funciones opcionales, recibe una lista delimitada por espacios, como:

careportal rawbg iob, consulte los complementos a continuación.

Este campo de ENABLE se puede dejar tal cual la Imagen 28 o se pueden colocar los siguientes parámetros[74]:

careportal openaps iob bwp cage basal pump bridge cod loop basal sage overridemaker viewing.

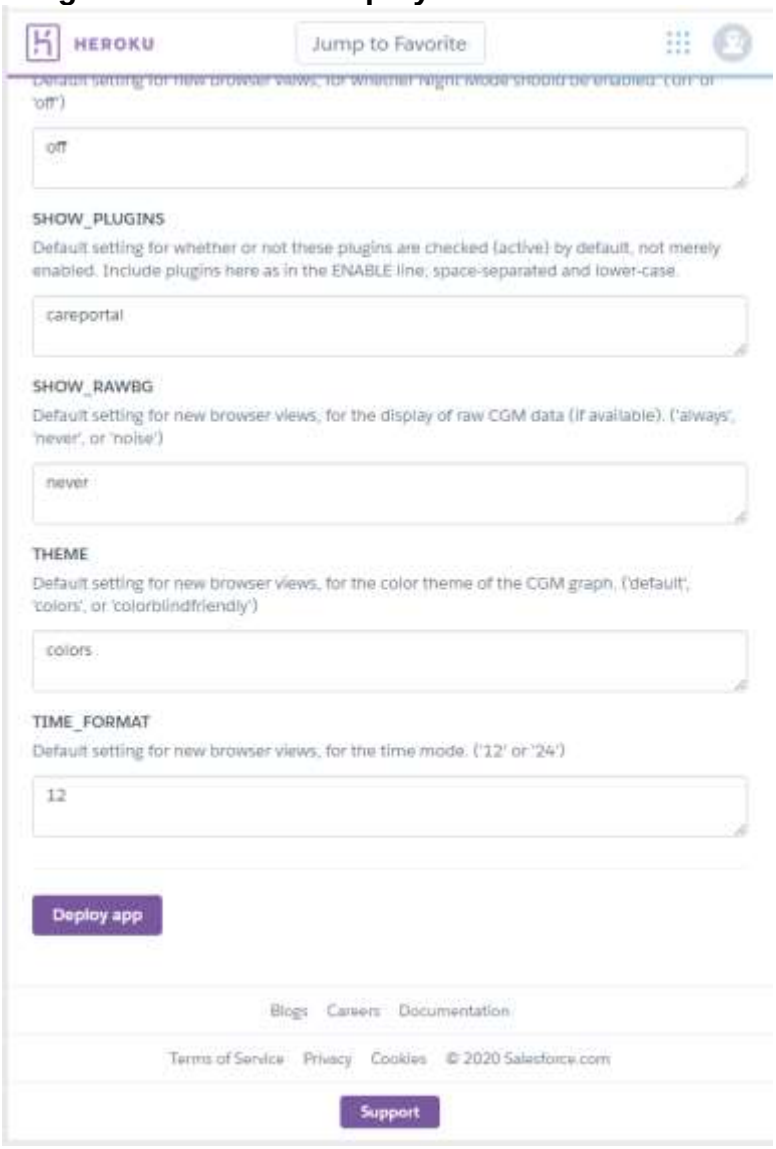
Estos parámetros son recomendados por la comunidad de Nightscout y sus usuarios. donde:

[74] Elements Marketplace: CGM Remote Monitor. {En línea}. {28 octubre de 2020} disponible en: https://elements.heroku.com/buttons/nightscout/cgm-remote-monitor

Careportal: Un formulario opcional para ingresar tratamientos.

Una vez realizado todos los pasos anteriores hay que dirigirse al botón “Deploy app” que se encuentra al final del formulario con el objetivo de empezar a migrar la app a Heroku e Implementar la App Web Nightscout (ver imagen 32).

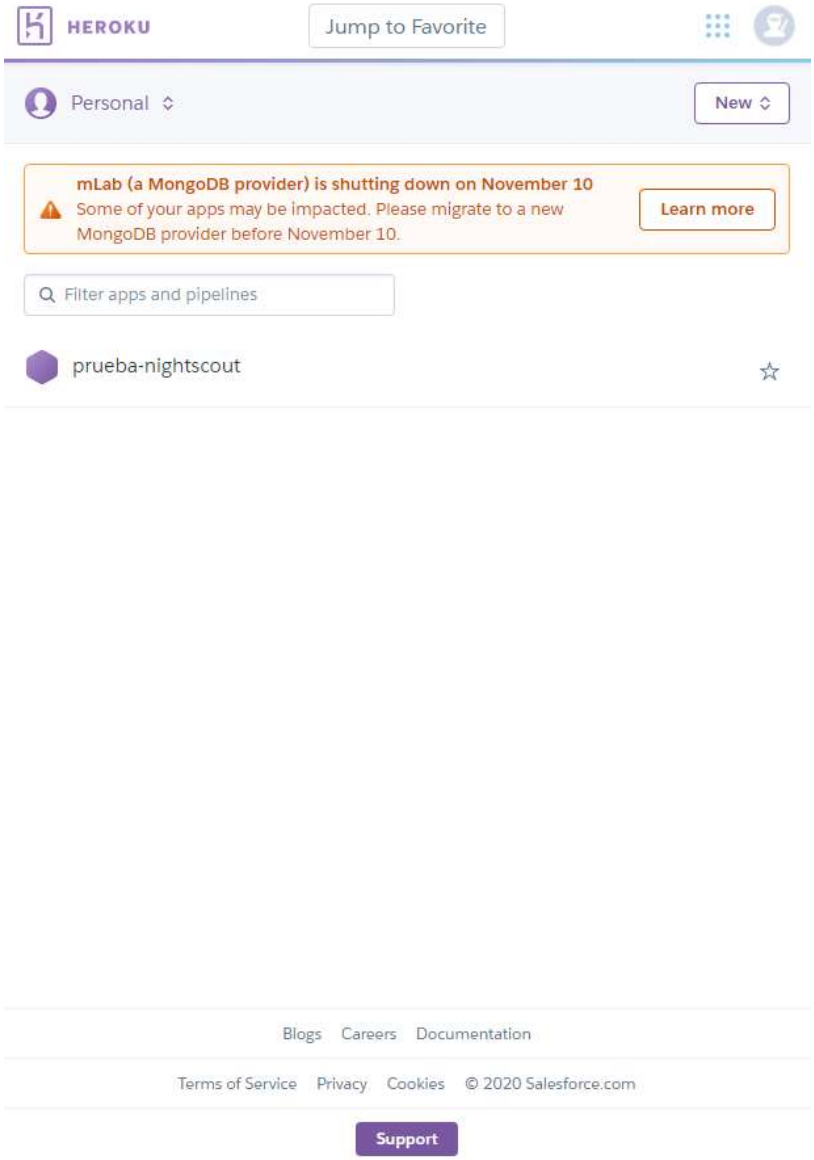
Imagen 32. Botón de Deploy Heroku.



Fuente: Autor desde dashboard Heroku.

Una vez se ha creado la aplicación en Heroku, continúa el proceso con el Deploy desde GitHub para ello se ingresa a la opción que tiene el nombre de la aplicación ver la imagen 33:

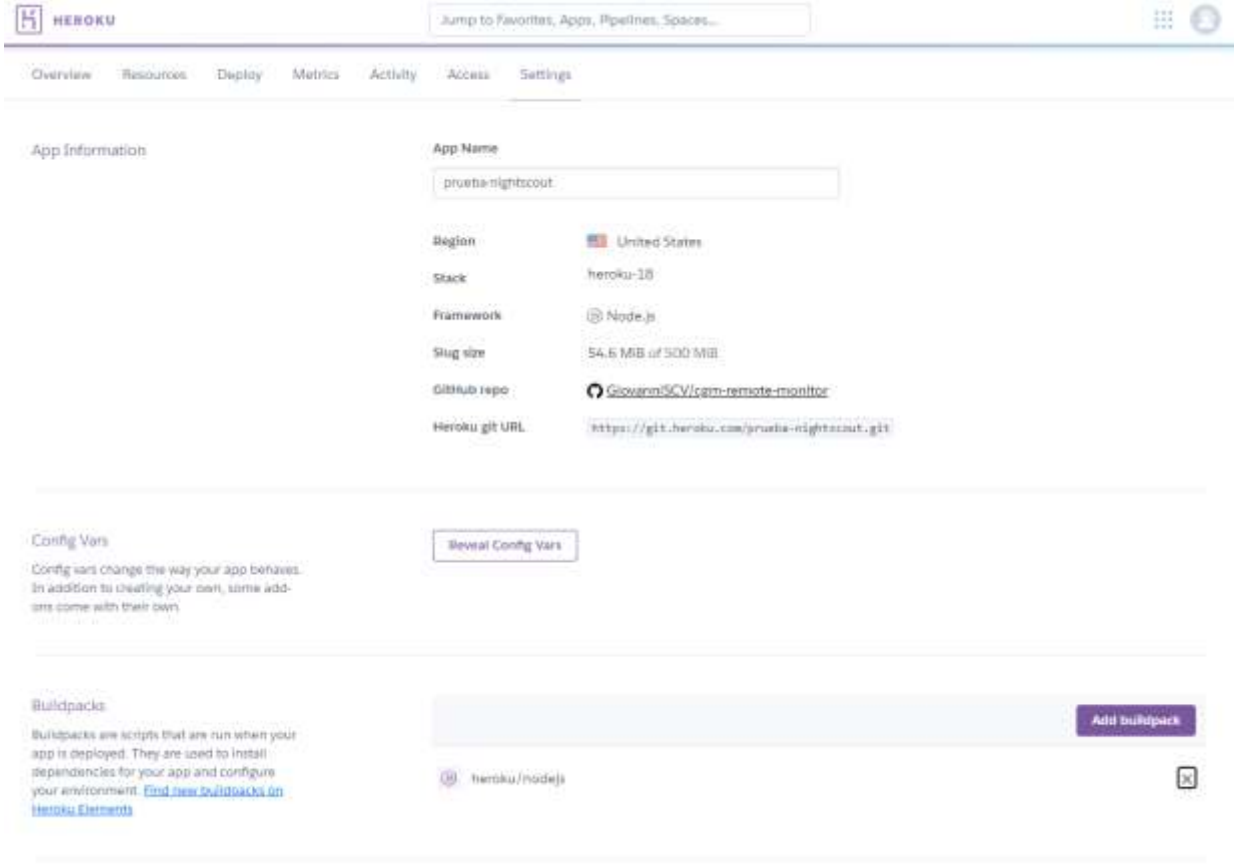
Imagen 33. Panel Inicial (Dashboard) Heroku.



Fuente: Autor desde dashboard Heroku.

Una vez se ha seleccionado hay que ingresar a la opción “Settings” que nos permitirá configurar el String que permitirá conectar la aplicación Nightscout en el servidor de Heroku con MongoDB Atlas, Al seleccionar la opción” Settings” da una vista general de las configuraciones como se muestra a continuación en la imagen 34. Y se debe seleccionar la opción “Reveal Config Vars”.

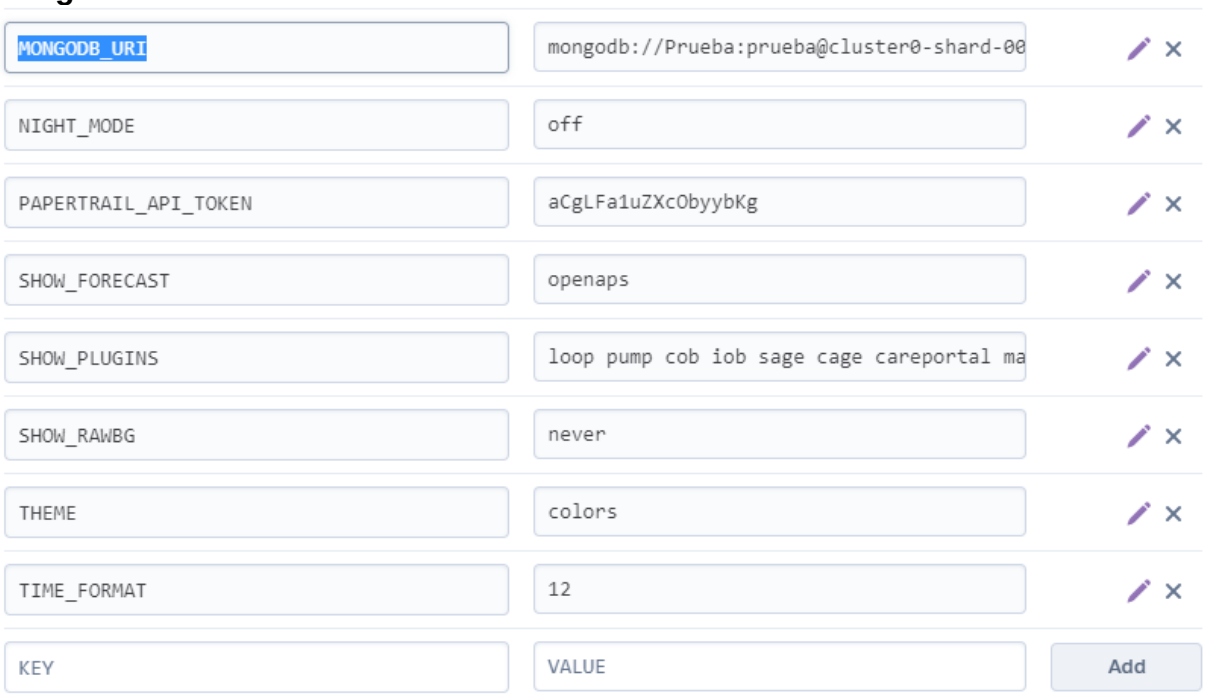
Imagen 34. Configuraciones y variables de Heroku.



Fuente: Autor desde dashboard Heroku.

Al seleccionar la opción “Reveal Config Var” hay una serie consecutiva de variables y su respectivo valor, estas variables son importantes para el funcionamiento correcto de la App Web Nightscout, para generar la conexión con MongoDB Atlas es necesario agregar una variable llamada “MONGODB_URI” y el string de conexión que nos brinda MongoDB atlas (Es necesario primero crear una cuenta en MongoDB atlas y seguir los pasos para crear un cluster para ahí sí tener un string de conexión) así como se muestra en la imagen 35.

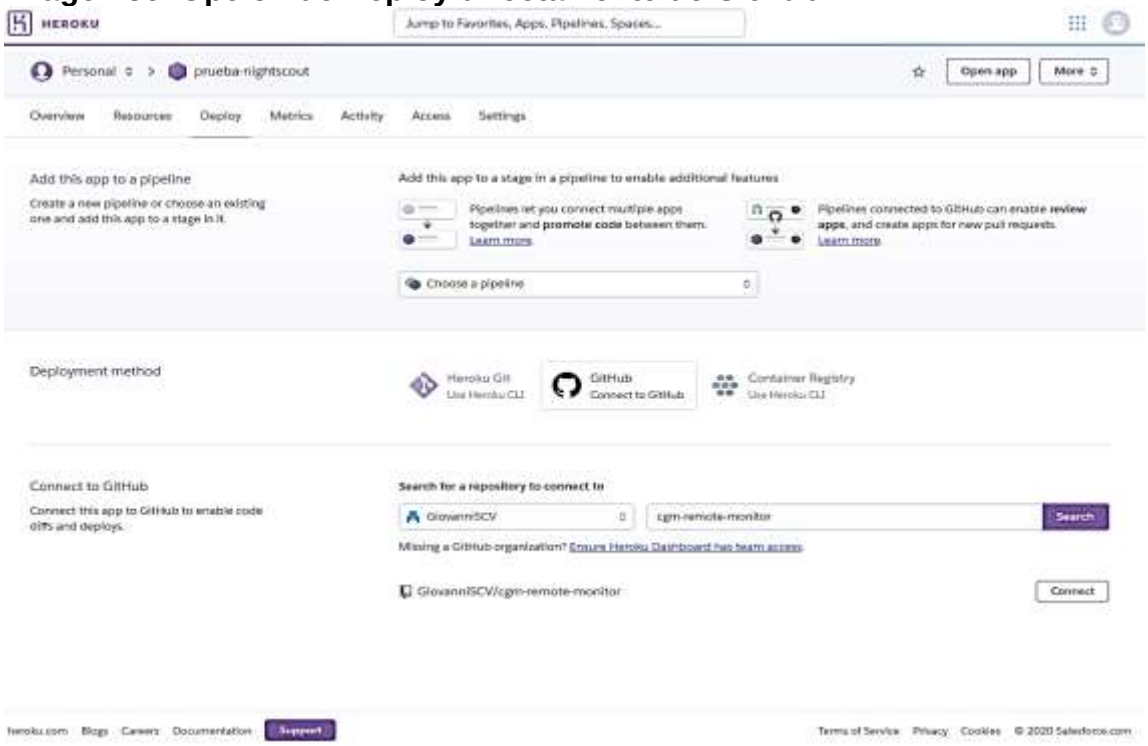
Imagen 35. Ejemplo de pareja Variable y su valor para generar conexión con MongoDB atlas.



Fuente: Autor.

Una vez se ha hecho el proceso anterior se debe ingresar a la opción “Deploy” que nos permitirá desplegar la aplicación en el servidor de Heroku, para este Deploy se encuentran 3 opciones, y para el caso de este trabajo de grado se seleccionó hacer el Deploy directamente de GitHub como se observa en la imagen 36.

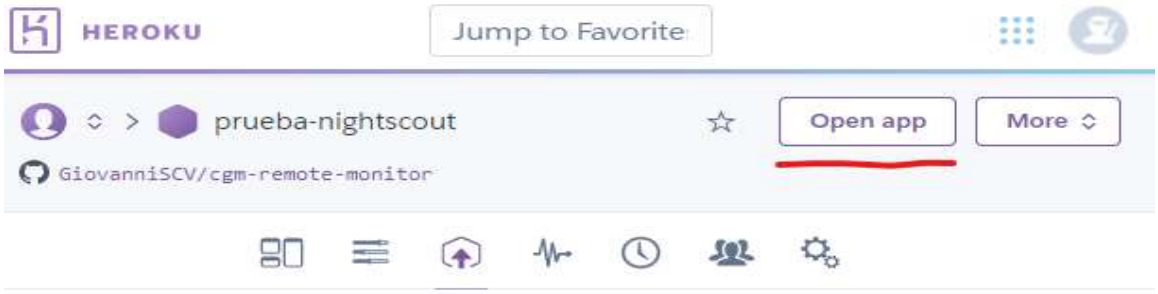
Imagen 36. Opción de Deploy directamente de GitHub.



Fuente: Autor.

Una vez se da click sobre el botón “Connect” solo resta darle al botón “Deploy” y se ejecuta e instala sola la aplicación de Nighscout, al finalizar en la parte superior de la barra de navegación de Heroku sale la opción de abrir la aplicación (ver imagen 37).

Imagen 37. Abrir y visualizar la aplicación desde Heroku.



Fuente: autor.

Una vez se da click en este botón se abrirá la aplicación como se muestra en la Imagen 38 que está a continuación.

Imagen 38.. Aplicación web Nightscout desplegada en Heroku.

Fuente: Autor.

Para este caso, no hay datos ingresados por el sensor porque no se ha conectado uno nuevo, pero una vez que está el sensor aplicado y funcionando con la aplicación Glimp en el caso de Android se puede visualizar en esta aplicación Web Nightscout. El siguiente paso consiste en la configuración de la aplicación Glimp para subir los datos a Nightscout.

6.2 SUBIDA DE DATOS DE FREESTYLE LIBRE A NIGHTSCOUT

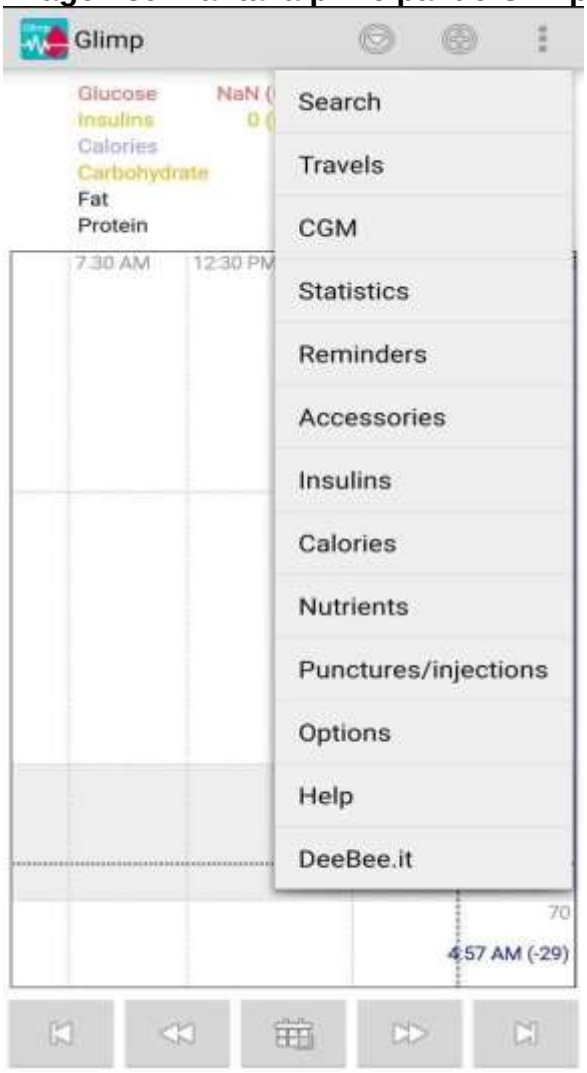
Para que el sitio Nightscout de este trabajo de grado obtenga datos, es necesario cargar(subir) los datos del sensor FreeStyle libre al servidor (la nube). Esto es posible mediante la aplicación de Glimp, la cual es una aplicación diseñada para el sistema operativo Android, esta aplicación es capaz de leer datos del sensor FreeStyle libre usando la tecnología NFC y subirlas a Nightscout. Para la instalación de Glimp para que funcione su conexión con Nightscout se realiza lo siguiente

Nota: La aplicación a la fecha en la que se realizó este trabajo de grado no está funcionando correctamente al 100% debido a que cuando uno realiza la prueba de

conexión genera un error (), pero realmente si hay conexión entre Glimp y el sitio web Nightscout.

1. Se debe descargar e instalar Glimp desde un lugar seguro como Google Play Store.
2. Se debe configurar Glimp para que funcione con su sistema FreeStyle libre. Luego se configura la subida de los datos a Nightscout de la siguiente forma:
    a. En la aplicación de Glimp seleccione los tres puntos en la parte superior derecha (que corresponde al menú de Glimp) y después seleccione opciones (Options) así como se muestra a continuación en la imagen 39.

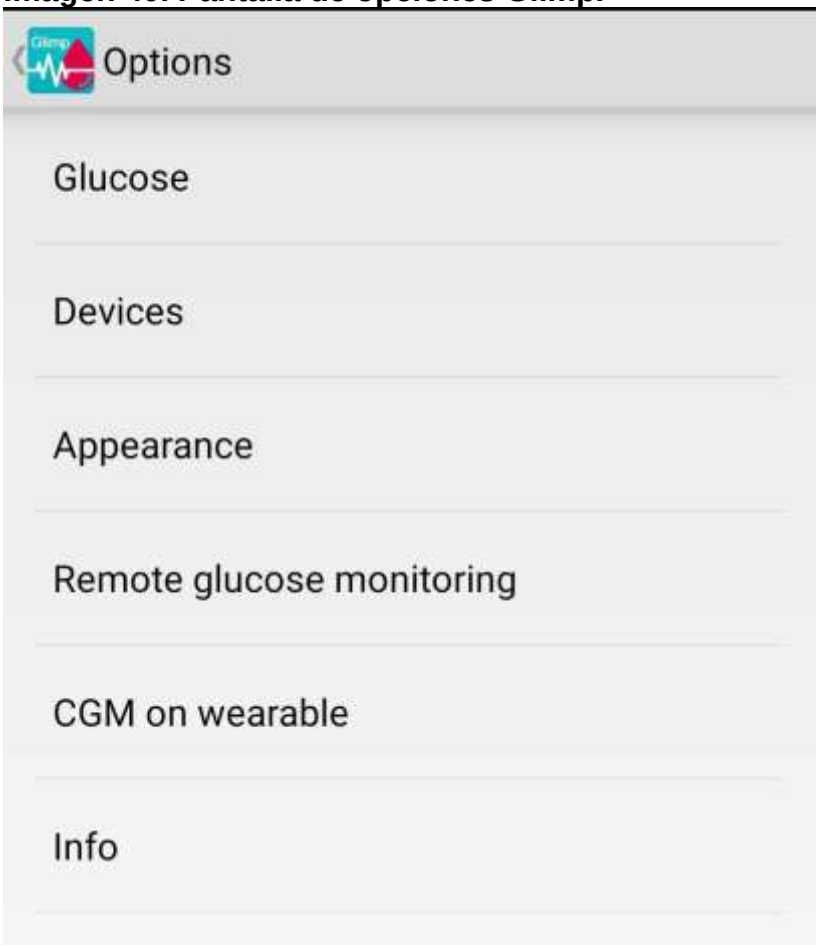
Imagen 39. Pantalla principal de Glimp, opciones.



Fuente: Autor.

b. Ahora seleccione Remote Glucose monitoring (Monitoreo remoto de glucosa) ver imagen 40.

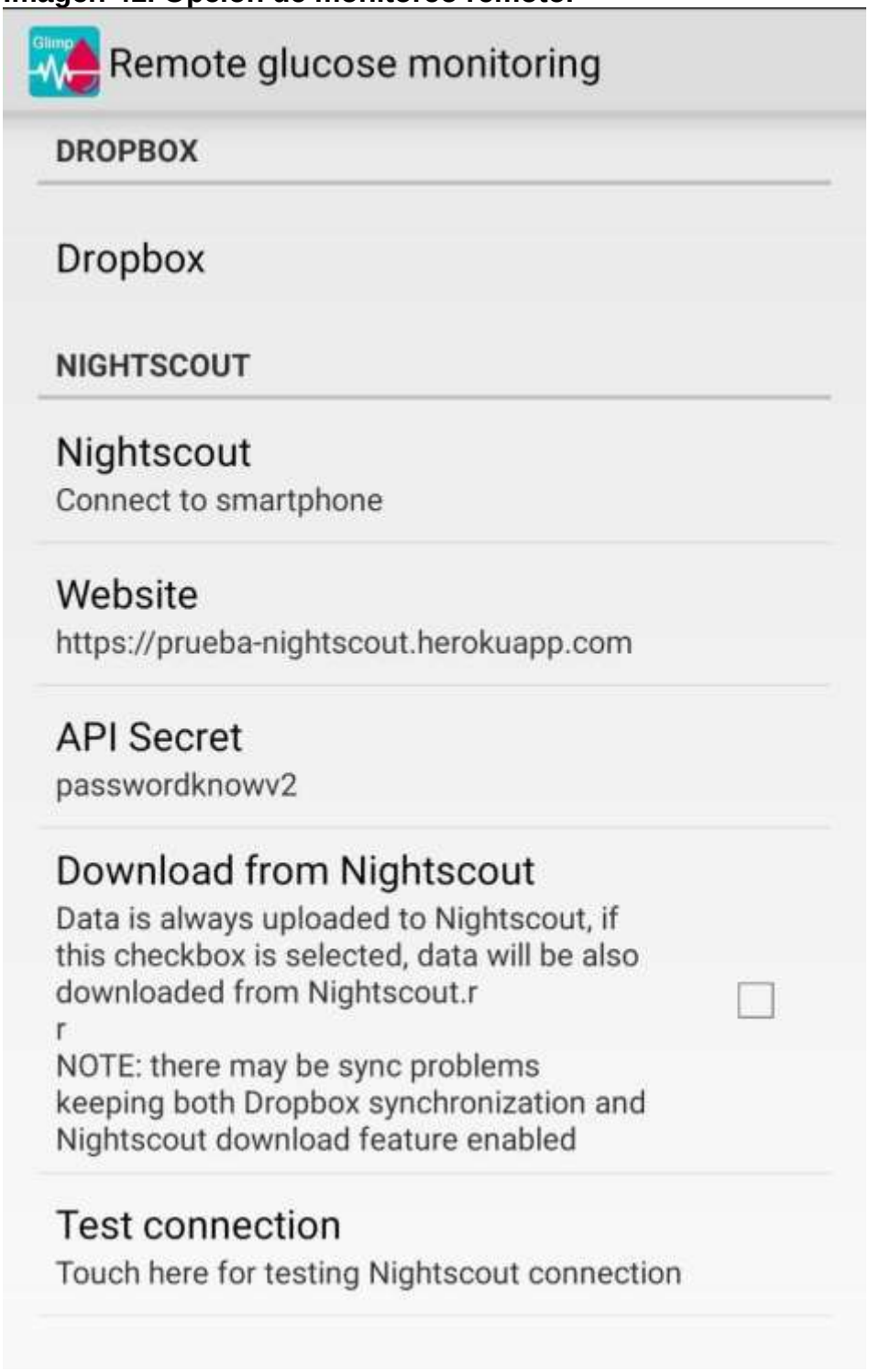
Imagen 40. Pantalla de opciones Glimp.



Fuente: Autor.

c. Dentro de Remote glucose monitoring, debe ingresar el URL de su Nightscout (ver imagen 41).

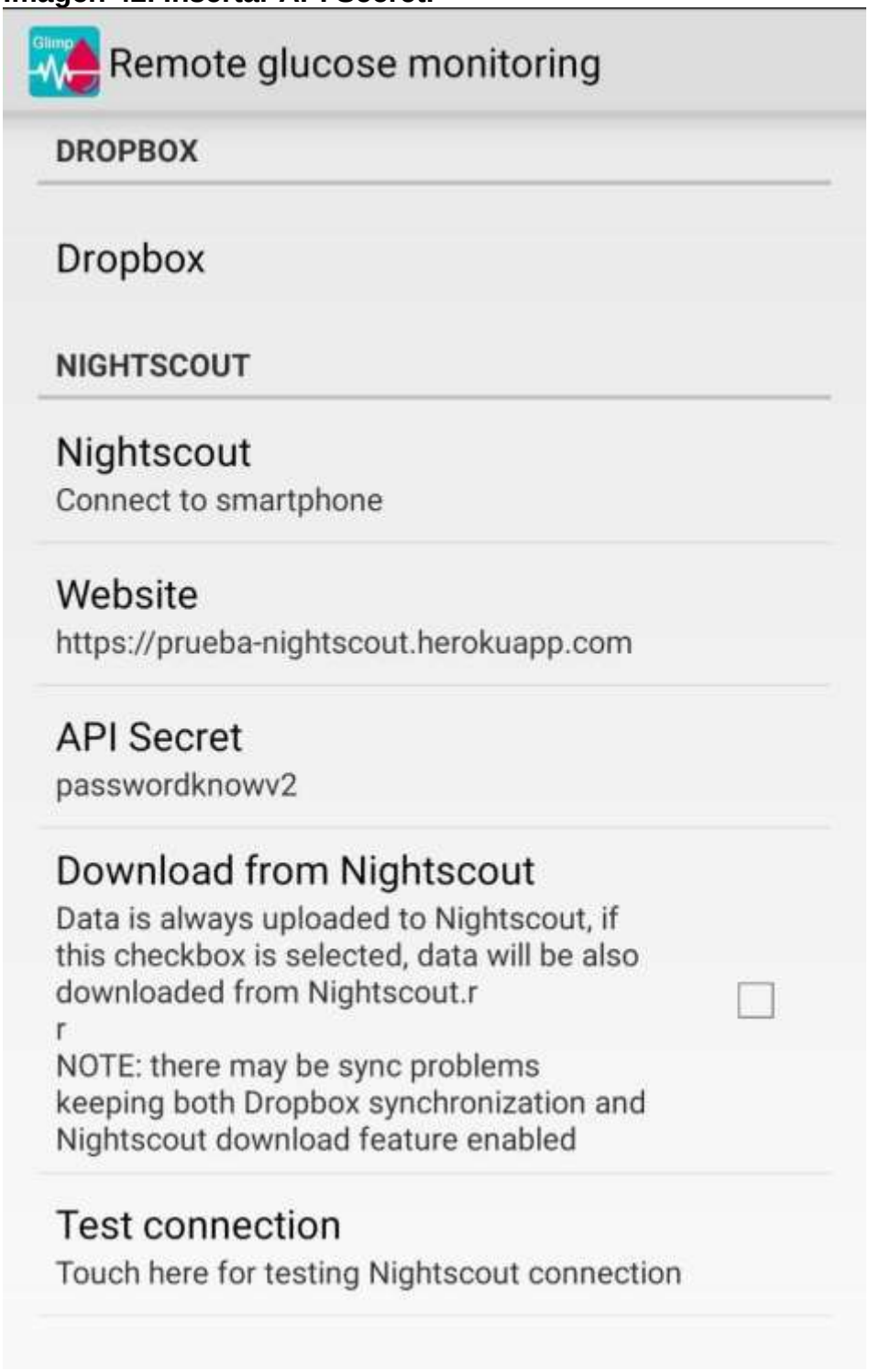
Imagen 41. Opción de monitoreo remoto.



Fuente: Autor

d. Después se debe ingresar el API Secret (ver imagen 42).

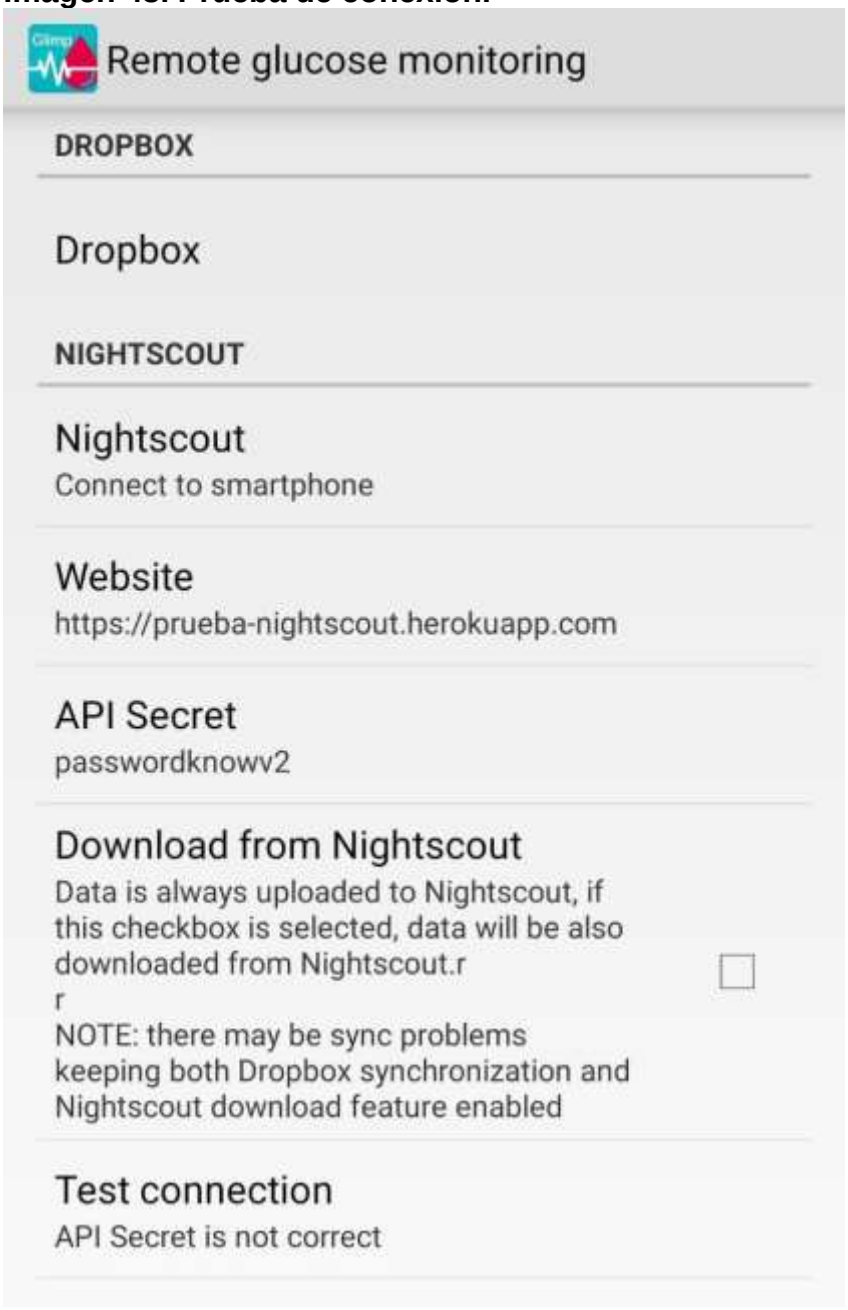
Imagen 42. Insertar API Secret.



Fuente: Autor.

e. Una vez ingresado el URL y el API Secret se realiza la prueba de conexión con el botón “Test connection” de la siguiente forma (ver imagen 43).

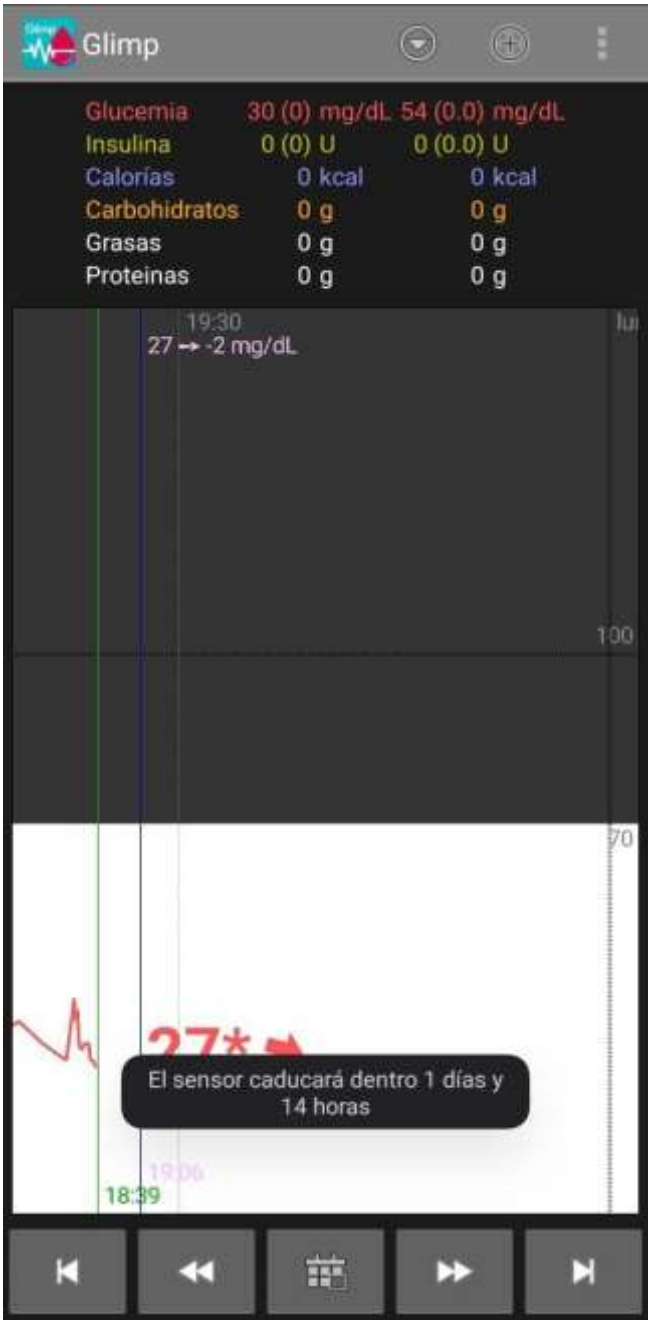
Imagen 43. Prueba de conexión.



f. Según la documentación de Nightscout, después de presionar en el botón de “Test connection” debería salir un mensaje de conexión

exitosa, para el caso de este trabajo de grado el mensaje que arroja es “API Secret is not correct”, que significa que la contraseña que se configuro en el Deploy es incorrecta, mensaje que es erróneo debido al hecho de que efectivamente sí se ha realizado una conexión con el servidor de Heroku (ver imagen 44).

Imagen 44. Lectura con aplicación Glimp del Sensor Freestyle Libre.



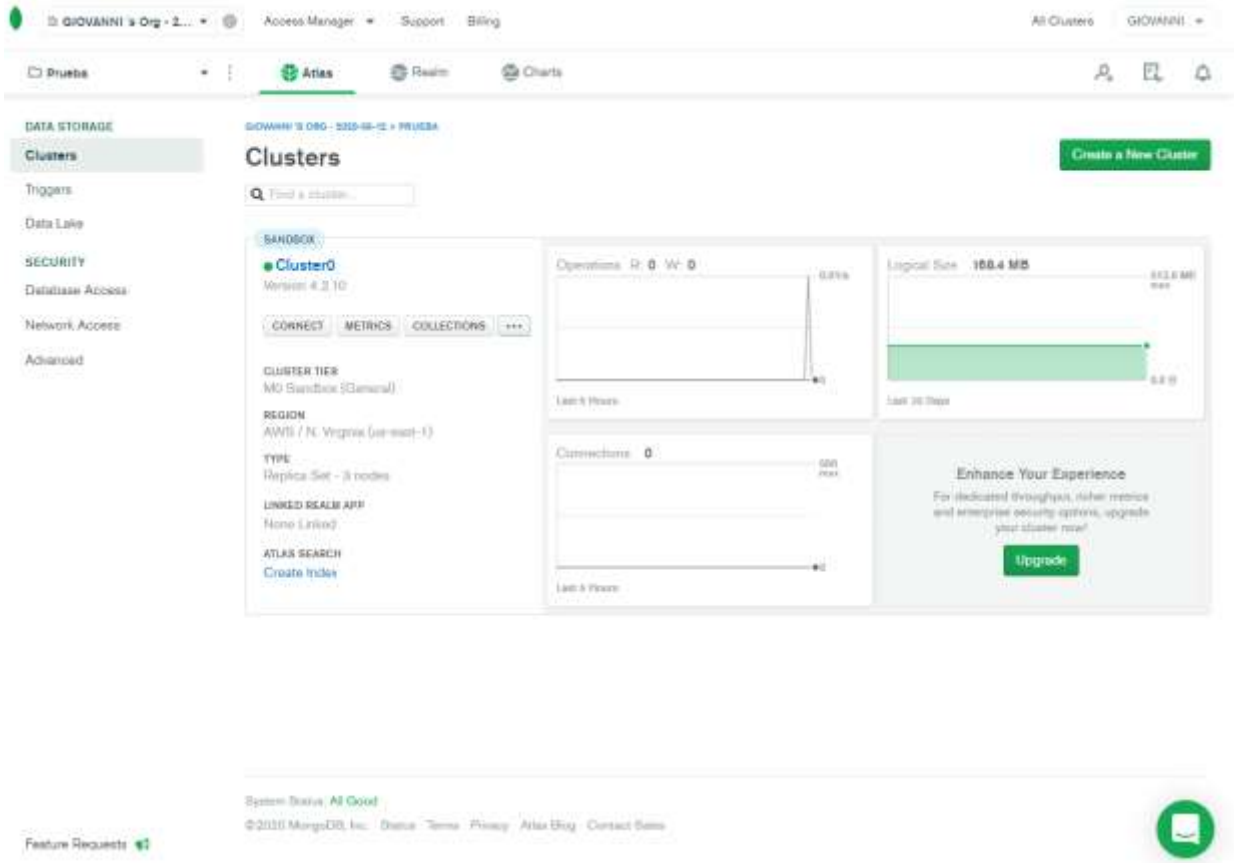
En la anterior imagen 44 se muestra como es la lectura de los datos desde la aplicación Glimp

6.3 SERVIDOR MONGODB ATLAS.

Para el funcionamiento correcto de la aplicación web brindada para la comunidad de Nightscout, es necesario tener una cuenta en el servicio ofrecido por MongoDB Atlas ya que esta aplicación necesita una base de datos no relacional en la cual almacenar los datos leídos del sensor FreeStyle Libre de Abbott por medio de la aplicación móvil Glimp para posteriormente visualizar la información contenida en la base de datos.

De acuerdo con lo anterior, es necesario tener una cuenta en el servicio ofrecido por MongoDB Atlas. Al tener una cuenta este servicio nos permite de forma gratuita genera un Clúster en el que se podrá generar una base de datos para este trabajo de grado se creó una llamada "**Prueba**", Una vez que se creó esta base de datos se procede a la página principal de MongoDB Atlas como se ve en la imagen 45.

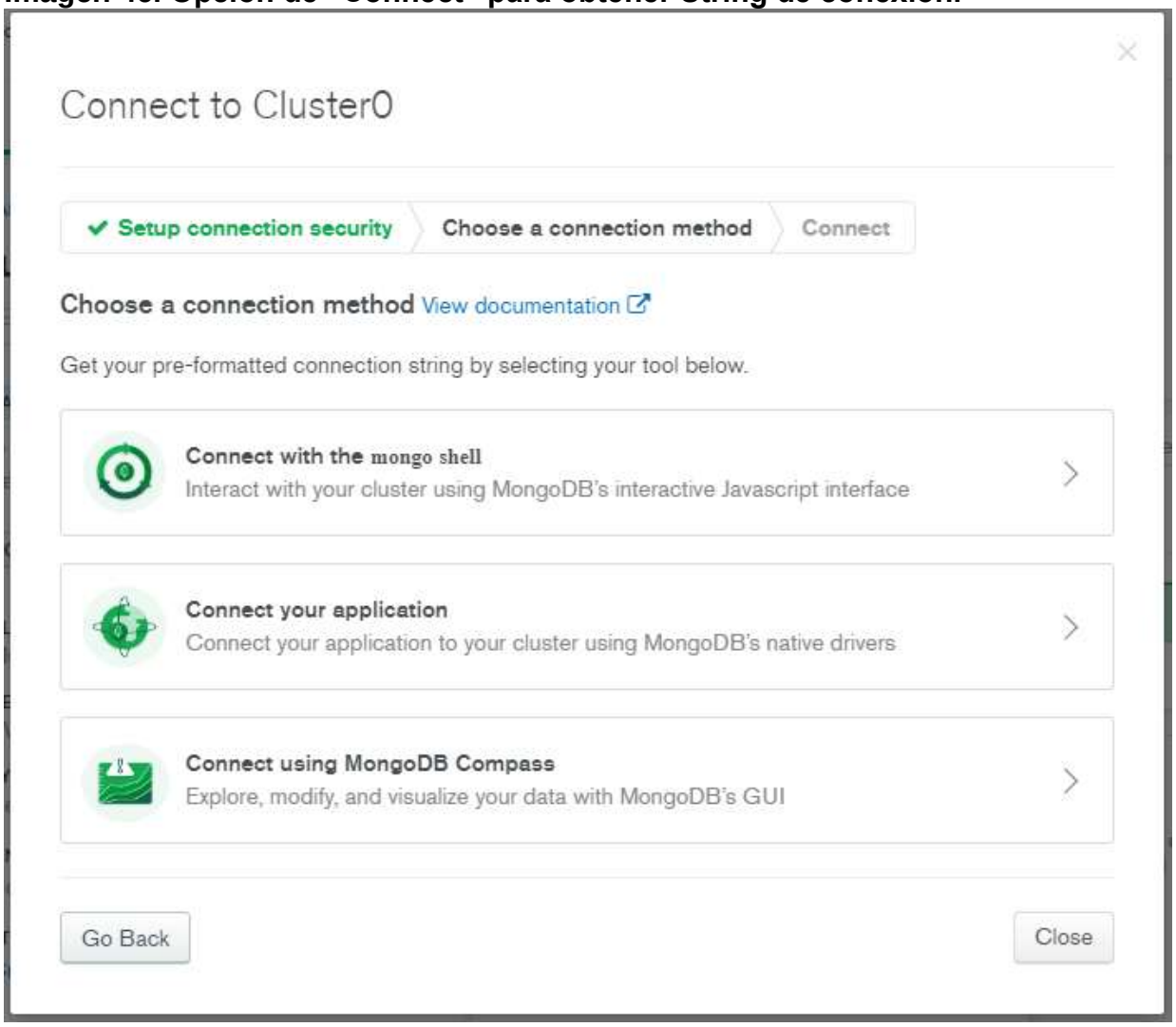
Imagen 45. Página principal de MoongoDB Atlas



Fuente: Autor.

En la página principal se observa la información del Cluster que este servicio nos da de forma gratuita, así como también algunas opciones como el acceder a las colecciones de las bases de datos, a las métricas y muy importante la opción de conexión en donde genera una cadena de conexión (String) que permite a los lenguajes de programación por medio de drivers acceder a la base de datos. El proceso para obtener esta cadena de conexión es muy sencillo y es como se muestra a continuación en la Imagen 46.

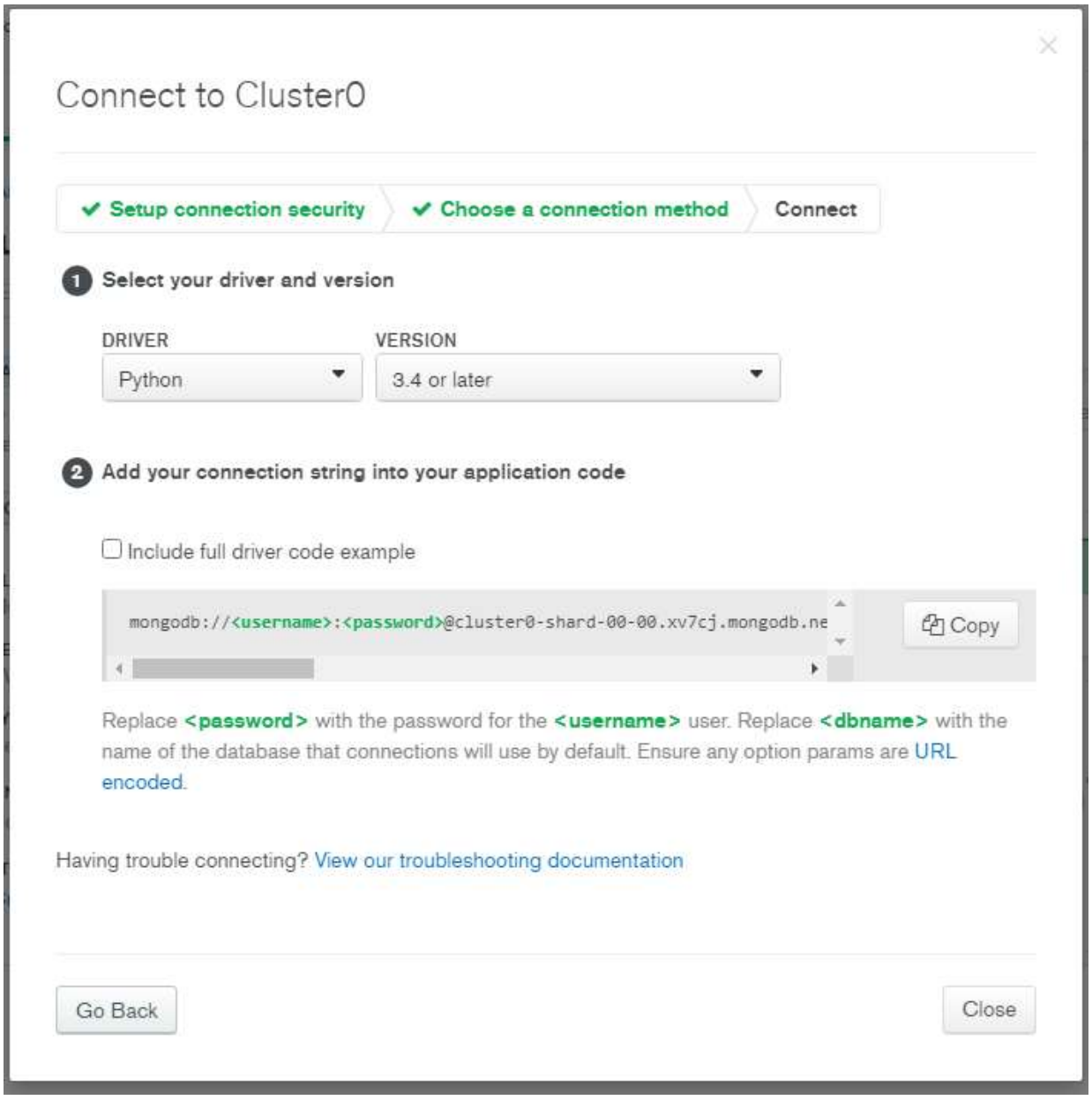
Imagen 46. Opción de "Connect" para obtener String de conexión.



Fuente: Autor.

Dentro de la ventana que se despliega al seleccionar la opción de "Connect" que se observa en la imagen 46 anterior, se encuentran tres opciones de las que para este trabajo de grado se seleccionó la opción "Connect your application", esta opción despliega una ventana con las opciones que se ven la Imagen 47.

Imagen 47. Ventana donde se genera el lengua y versión para String de conexión.



Fuente: Autor.

En las opciones que se observan en la Imagen 47, primero se debe seleccionar el driver relacionado con el lenguaje y la versión, para el Deploy de la aplicación web de Nightscout se selecciona el driver de node.js (this is JavaScript Runtime) para versiones 3.4 o superior, debido a que esta aplicación funciona en gran parte a el lenguaje de JavaScript, por otro lado, también de se seleccionó después la opción del driver del lenguaje Python con una versión 3.4 o superior. cualquiera de las dos

opciones brinda los siguientes String que se utilizaran para los drivers de conexión de cada uno de los lenguajes correspondientes.

String para driver node JS.
mongodb+srv://<username>:<password>@cluster0.xv7cj.mongodb.net/<dbname>?retryWrites=true&w=majority

Stringo para driver Python.
mongodb+srv://<username>:<password>@cluster0.xv7cj.mongodb.net/<dbname>?retryWrites=true&w=majority

Como se puede observar en cada uno de los dos string de conexiones anteriores, aparte de que son iguales, consta de campos para agregar el usuario, la contraseña del usuario y la base de datos que como se mencionó anteriormente para este trabajo de grado se llama Prueba.

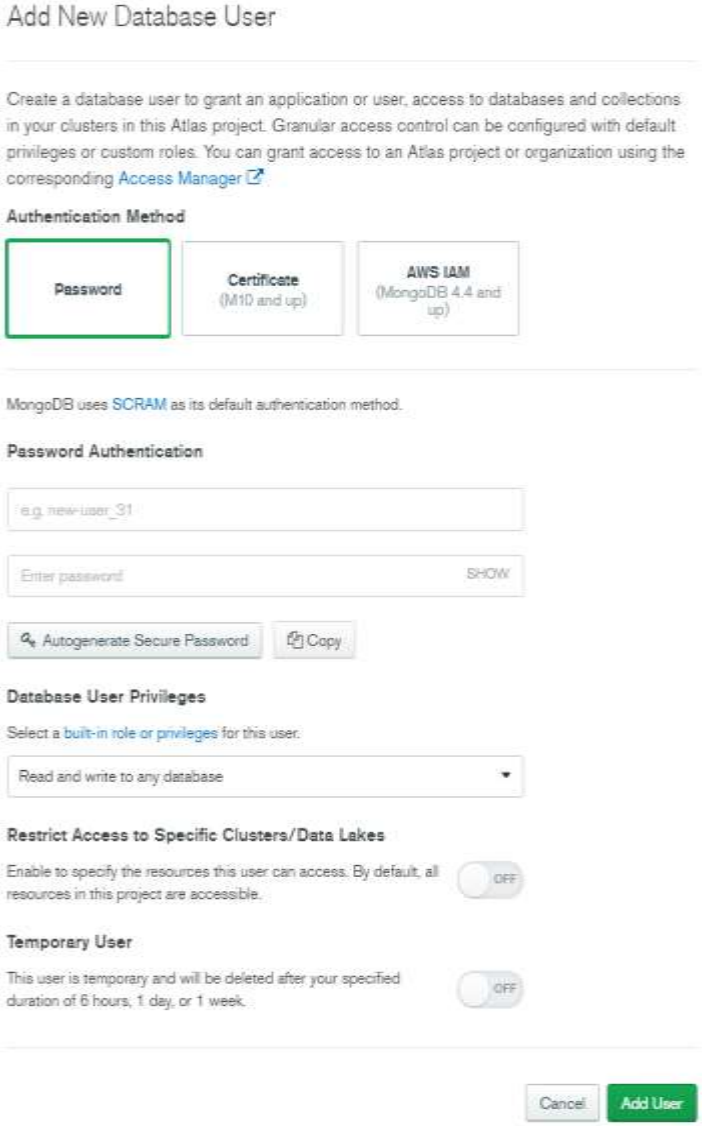
Para obtener el usuario y la contraseña que son necesarios para la conexión es necesario crear estas credenciales, por lo que se debe ir a las opciones de la izquierda en donde se pueden hacer configuraciones del servicio y entre ellas se pueden agregar credenciales de conexión de usuario para las bases de datos, darle permisos ya sea solo de lectura, lectura y escritura o de administrador, dependiendo de la necesidad, para acceder a estas opciones se selecciona como se ve en la Imagen 48.

Imagen 48. Opciones que brinda MongoDB Atlas.

Fuente: Autor.

De las opciones que se mostraron anteriormente para el acceso a las bases de datos disponibles, se selecciona la opción de “Database Access” en donde despliega la ventana que se muestra en la imagen 49 en donde se ingresan los parámetros de nombre de usuario y contraseña, así como también se selecciona el método de autenticación (de forma gratuita esta Authentication Method), también los privilegios del usuario de la base de datos en donde se seleccionó de lectura y escritura para cualquier base de datos dentro del Cluster.

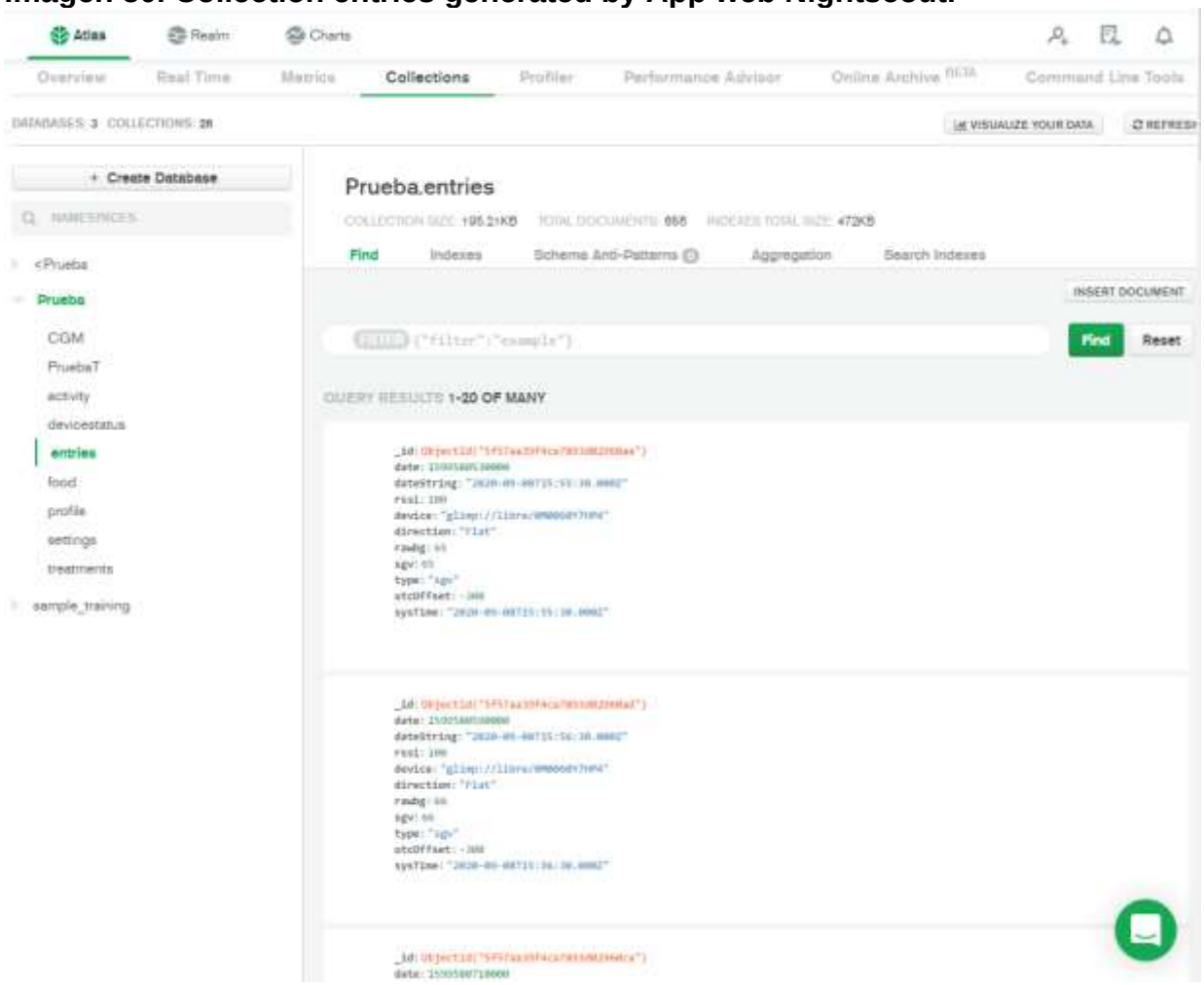
Imagen 49. Creación de Usuario y contraseña de Base de Datos.



Fuente: Autor.

Con esto ya se tiene todo lo necesario para poder generar una conexión al Cluster y así como a todas las bases de datos dentro de él. y una vez que se realizan los pasos del Deploy de la aplicación de Nightscout con el string que se generó en los pasos anteriores, se observa que la aplicación genera una colección de la siguiente forma (ver imagen 50):

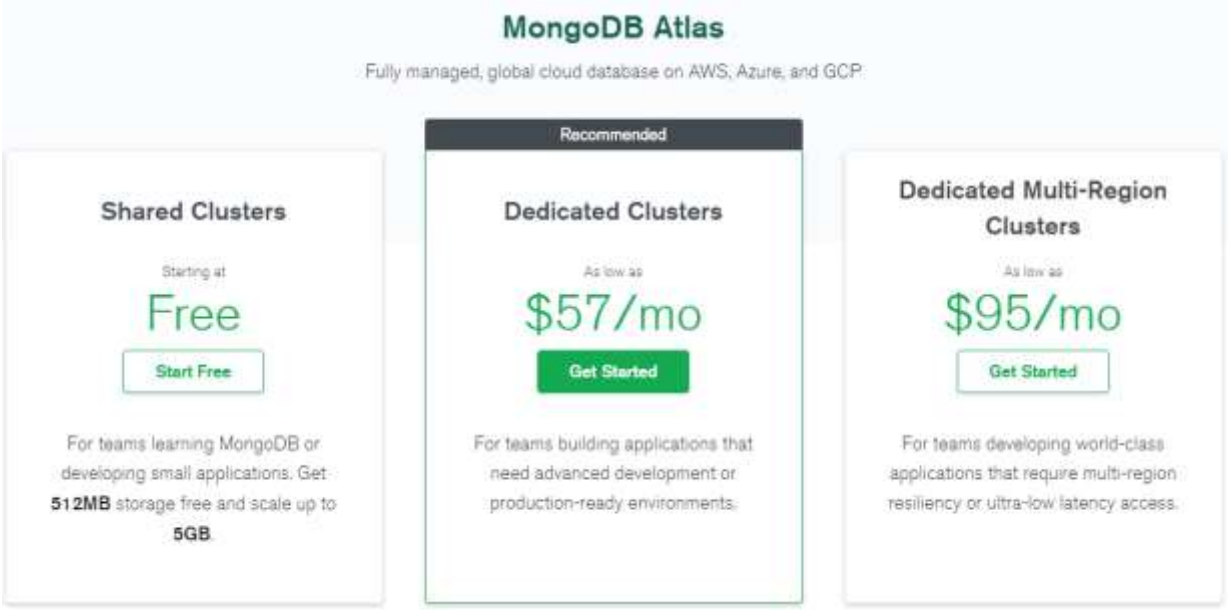
Imagen 50. Collection entries generated by App web Nightscout.



Fuente: Autor.

Una vez ya está en funcionamiento la base de datos MongoDB, se procede a crear un Add-ons en Heroku, para crear una base de datos de prueba MySql. El Add-ons que se seleccionó es ClearDB (ver imagen 51), el cual permite generar bases de datos MySql de prueba de forma gratuita con un almacenamiento máximo de 5 MB, y 10 conexiones simultáneas entre otras características.

Imagen 51. Plan de precios MongoDB Atlas.



Fuente: MongoDB

La característica de Add-ons de ClearDB, se instala en el espacio en memoria reservada para la aplicación que se subirá a este servidor con el objetivo de recibir los valores enviados por la báscula de usuario, almacenarlos en la base de datos MySql también esta aplicación se debe encargar de hacer el traspaso de los datos almacenados en la base de datos MongoDB a MySql. ClearDB Brinda acceso el servidor MySql y el string de conexión que se utilizara con el driver de la aplicación desarrollada en Flask micro Framework de Python.

Según la documentación propia de Flask es autodenominado microframework lo que significa que Flask tiene como objetivo mantener una composición simple, pero con la posibilidad de extender funciones por medio de módulos del propio framework[75]. Flask a diferencia de un framework como Django, no toma decisiones por el desarrollador lo que permite que el desarrollador tenga una mayor comprensión de lo que está incluyendo en su proyecto.

Esto quiere decir que da libertad para la selección de base de datos a utilizar sin preconfiguraciones de alguna otra. por defecto el microframework Flask no incluye funcionalidades de abstracción de bases de datos o validación de formularios o algo similar donde ya existen bibliotecas diferentes que puedan ejercer esta función. En vez de esto lo que hace Flask es que admite extensiones que se integran en el proyecto de forma tal como si se hubiera implementado en el propio Flask. Entre las extensiones que son útiles para este trabajo de grado se encuentran las extensiones

[75]Flask {En línea}. 2010 {25 de octubre 2020} disponible en: https://flask.palletsprojects.com/en/1.1.x/

para conexión con las bases de datos, validaciones de formularios (Para recibir los métodos POST y GET), uso de hilos para emular acciones en paralelo entre otras extensiones[76]. Para la estructura del proyecto Flask que se desarrolla para este trabajo de grado, consta de los siguientes componentes:

6.4 DESARROLLO APLICACIÓN WEB CON FLASK.

Componentes

- archivo app.py
- carpeta static
- carpeta templates
- archivo gitignore
- Procfile.
- Requirements.

La arquitectura inicia con el archivo app.py el cual es un archivo de extensión Python, y dentro de este se asigna toda la lógica de negocios de la aplicación web. La lógica implementada dentro del archivo .py consta de:

- Llamado e importación de extensiones necesarias para el funcionamiento adecuado del proyecto. Aquí se importa principalmente el módulo de Flask.
- luego de se escriben los comandos que permiten generar la aplicación, estos comandos son los recomendados por la documentación propia de Flask.
- Luego va la asignación de los string que se van a utilizar para la conexión con las bases de datos y se inicializa la conexión con MySql y MongoDB.
- Se genera la función “traspaso_datos ()” la cual no recibe nada como parámetro, y dentro de esta función se encuentra la lógica que se encarga de generar la conexión con la base de datos MongoDB leer los datos, y prepararlos para que posteriormente sean insertados en la base de datos MySql.
- para esta aplicación se decidió hacer uso de las características de los hilos que posee un procesador (en el servidor) y ejecutar esta función en segundo plano para que a su vez esta aplicación pueda recibir en todo momento (sujeto a la disponibilidad del servidor debido a que se adquirió un plan gratuito) los datos enviados por el usuario de su peso.
- El Trabajo de grado cuenta con un proyecto de software con las funcionalidades de enrutamiento presentes en toda aplicación web, estando relacionada a su vez esta ruta con su método. por lo que se tiene una ruta específica para el Home de la aplicación, y por otro lado una ruta encargada específicamente de recibir por los métodos

[76] Flask documentation. {En línea}. 2010 {25 octubre 2020} disponible en: https://flask.palletsprojects.com/en/1.1.x/foreword/#configuration-and-conventions

POST y GET los datos enviados de la báscula al dispositivo móvil, y del dispositivo móvil al servidor.

- Por último, se le indica que la aplicación se ejecuta como aplicación principal dentro del proyecto creado para este trabajo de grado, ya que los proyectos pueden estar compuestos por múltiples aplicaciones. y también mientras se hacían las pruebas se le decía que debía estar en modo debug para ir detectando errores, pero esta opción se debe desactivar una vez se tiene la versión completa y se sube a un servidor para generar la aplicación en producción por seguridad.

El archivo app.py se encuentra entre la carpeta raíz, es decir está en la carpeta principal del proyecto, después se encuentran dos carpetas dentro de esta carpeta principal, estas son “static” y “templates”. La carpeta static va a contener todos los archivos estáticos como pueden ser imágenes, JavaScript o css, y se dejan dentro de esta carpeta static para poder acceder a estos archivos fácilmente. Por otro lado, está la carpeta “templates” en donde se almacenarán todos los archivos HTML que se van a utilizar para la visualización del usuario, en donde se encuentra el archivo index.html para el caso de este trabajo de grado en donde se dará aviso del funcionamiento del servidor.

a continuación, se explica los archivos .gitignore, Procfile y requirements.txt, estos archivos son importantes para que el servidor Heroku identifique la aplicación como una aplicación Python[77]:

- .gitignore: consiste en un archivo que se encarga de decirle que a git que no queremos que rastree el contenido de lo que se encuentra de los paquetes que ahí se señalan.
- Procfile: es un archivo que se encuentra en la raíz de la aplicación y sirve para decir explícitamente el comando que se debe ejecutar al iniciar el servidor. Dentro de este documento se le asignó el comando para que la aplicación en Heroku reciba tráfico Web, debido a que Heroku lo agrega a su pila de enrutamiento HTTP y esto se ejecuta cuando se realiza el Deploy de la aplicación[78].
- Requirements.txt: los archivos que se encuentran dentro del archivo txt contienen un listado de las extensiones o dependencias necesarias para que se ejecute la aplicación correctamente. el servidor Heroku lo que hace es tomar este archivo al momento del Deploy y empieza a instalar todo lo mencionado en este archivo.

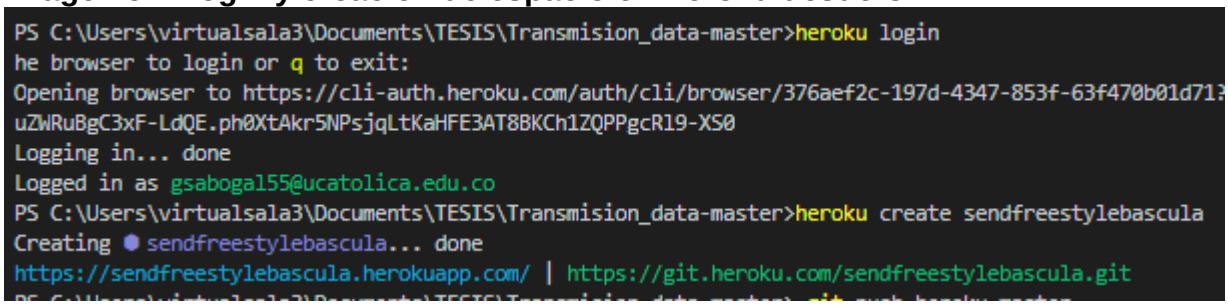
[77] Implementación de aplicaciones Python y Django en Heroku {En línea}. {25 octubre 2020} disponible en: https://devcenter.heroku.com/articles/deploying-python

[78] Getting Started on Heroku with Python {En línea}. {25 octubre 2020} disponible en: https://devcenter.heroku.com/articles/getting-started-with-python?singlepage=true

En conclusión, con todo lo anterior se tiene todo lo necesario para hacer el Deploy en Heroku para el funcionamiento de la aplicación y la integración de los datos. El proceso que se muestra a continuación es el que se lleva a cabo para subir la aplicación al servidor Heroku, esto se puede hacer una ver se ha descargado e instalado los CLI GIT y CLI de Heroku, en donde el CLI de Git permite subir y sincronizar el repositorio local, es decir la carpeta de la aplicación con la cuenta que se tiene creada en GitHub, para realizar un control de versiones, a su vez con la extensión del CLI de Heroku, se hace de forma sencilla el Deploy en este servidor una vez ya se tiene configurado todos los componentes explicados anteriormente.

Inicia el proceso del Deploy con la creación de un espacio dentro del servidor de Heroku para la aplicación que se desea subir.

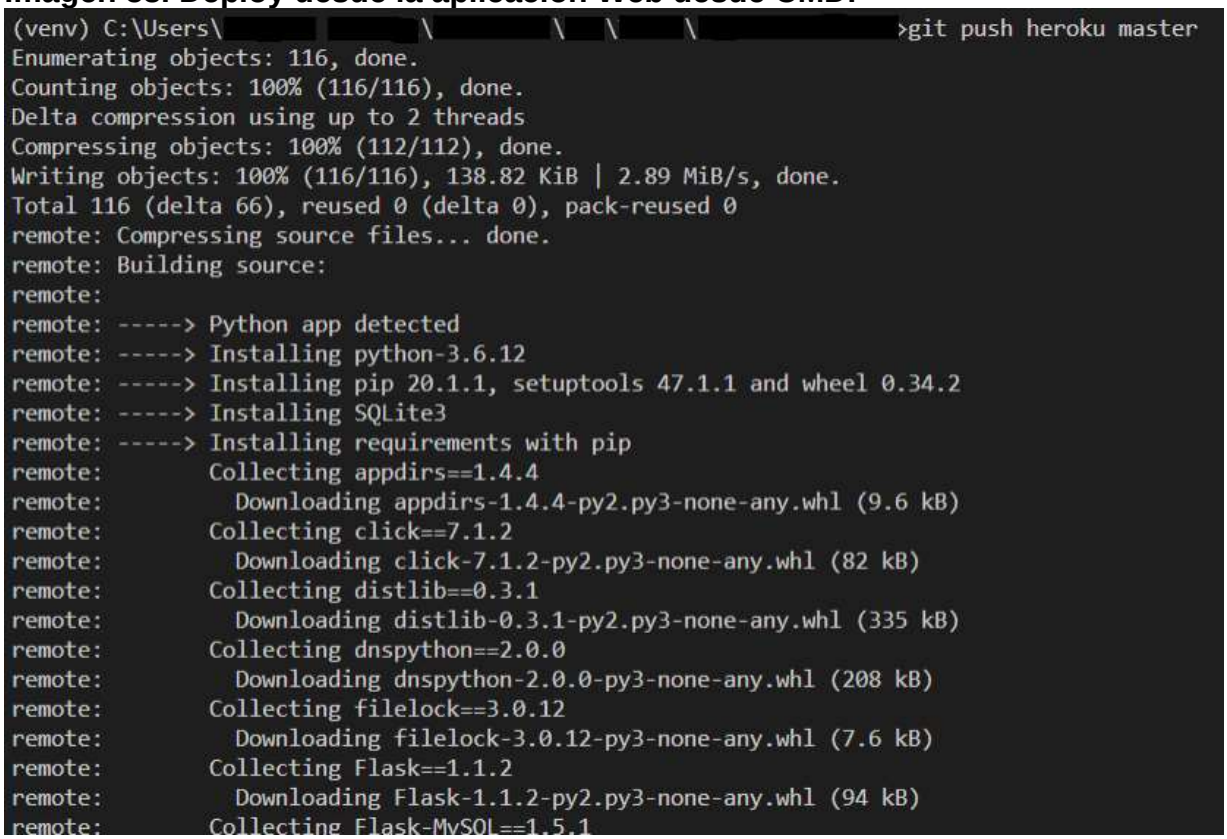
Imagen 52. Login y creación de espacio en Heroku desde CMD.



Fuente: Autor.

En la imagen 52 se observa los comandos necesarios para generar un login desde consola (ejemplo en cmd) y cómo crear un espacio para en el servidor Heroku para poder realizar el Deploy y asi poder subir la aplicación posteriormente en ese espacio dentro del servidor de Heroku.

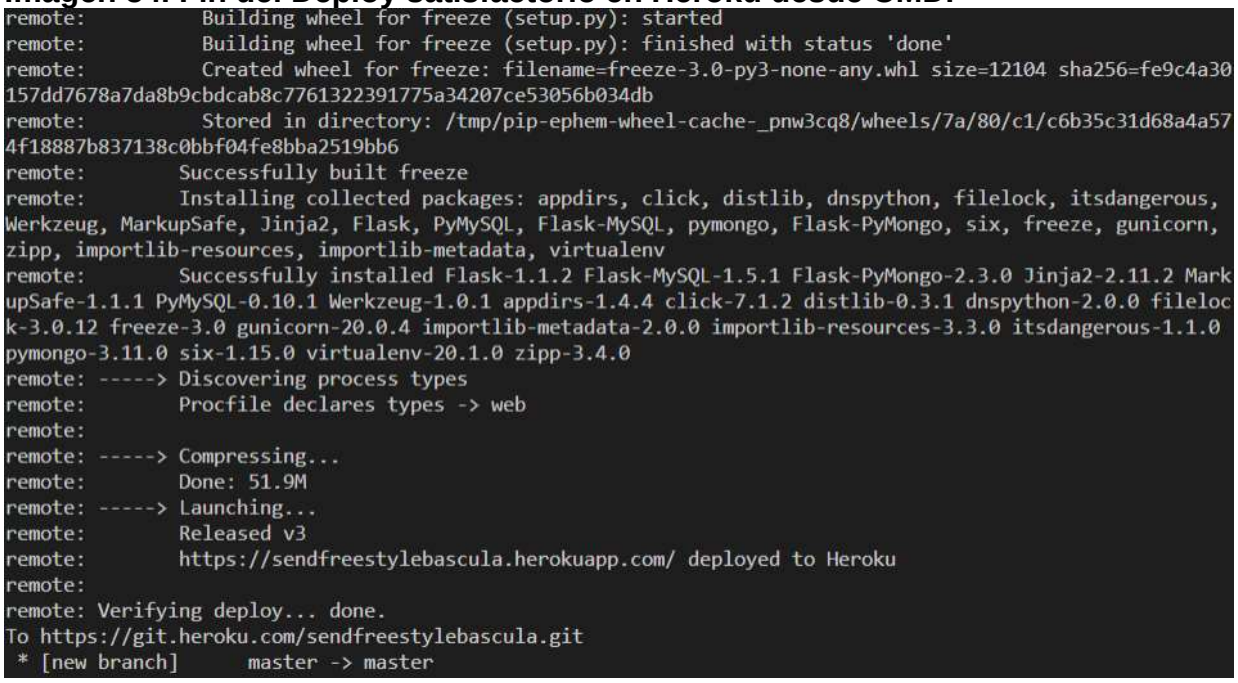
Imagen 53. Deploy desde la aplicación Web desde CMD.



Fuente: Autor.

Como se puede observar en la imagen 53. se inicia el proceso de instalación de todas las librerías necesarias que están contenidas en el archivo requirements.txt, así como los todos los demás archivos que componen la aplicación en el servidor de Heroku.

Imagen 54. Fin del Deploy satisfactorio en Heroku desde CMD.



Fuente: Autor.

En la imagen 54, se observa que el proceso de Deploy en Heroku ha terminado satisfactoriamente y ya está funcionando la aplicación, con esto termina el proceso de desarrollo e implementación de la aplicación que permita recibir y almacenar los datos tanto del sensor FreeStyle Libre de Abbott y la báscula. para finalizar se sigue con el proceso.

6.5 IMPLEMENTACIÓN BÁSCULA PARA ADQUISICIÓN DE VARIABLE DE MASA CORPORAL

Para la implementación del sistema de báscula consta de los siguientes elementos:

- 4 celdas de carga.
- Un módulo HX711.
- un módulo Bluetooth.
- Aplicación hecha en APP Inventor.

Una celda de carga según Moisés Espinosa, es un transductor electrónico que cumple la funcionalidad de transformar o trasladar fuerza o peso a variaciones de voltaje [79]. Estas variaciones producen en la instrumentación de salida una

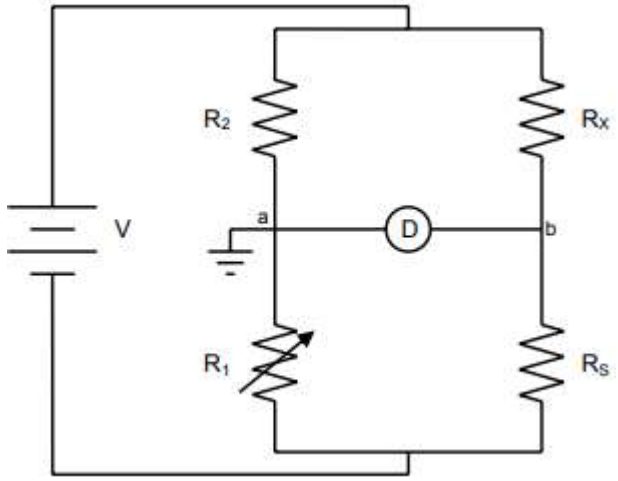
[79] ESPINOSA ESQUIVEL, MOISES. “DISEÑO y CONSTRUCCIÓN DE UNA CELDA DE CARGA”. SAN NICOLÁS DE LOS GARZA, N. L, 1995, 126p. Trabajo de investigación (grado de maestro en

desviación lineal, lo que permite que se pueda realizar calibraciones relacionadas con el peso aplicado directamente sobre las celdas de carga.

Las celdas de carga generan cambios lineales en su resistencia al variar el peso aplicado en ellas, el inconveniente con estas variaciones es que son relativamente pequeñas llegando a tener variaciones de 0.12 Ω si hablamos de que la resistencia total de esa celda de carga es de 120 Ω en total.

Lo que significa que para medir estos pequeños cambios en la resistencia se necesitan dispositivos de instrumentación muy precisos y costosos u otra opción es utilizar él puede Wheatstone, este circuito simple es capaz de percibir estas pequeñas variaciones de resistencia y entregar este valor en variaciones de Voltaje entre los puntos a y b[80], como se observa en la imagen 55. Los valores son captados gracias a una técnica conocida como equilibrio de brazos.

Imagen 55. Esquema Puente Wheatstone.



Autor: Puente de Wheatstone Modificado Utilizado en una Comparación Internacional en el Valor de 1 GΩ.

Conociendo la tensión de la fuente entregada por el Arduino para este trabajo de grado que corresponde a 5 V, y la tensión percibida en D (Tensión entre los puntos a y b), que es proporcional a la deformación que sufren las celdas de carga, donde de acuerdo a la imagen del esquema del puente Wheatstone, las cuatro celdas de

ciencias de la Ingenieria mecanica con especialidad en diseño mecanico). Universidad Autónoma de Nuevo León. Facultad de ingeniería mecánica y eléctrica.

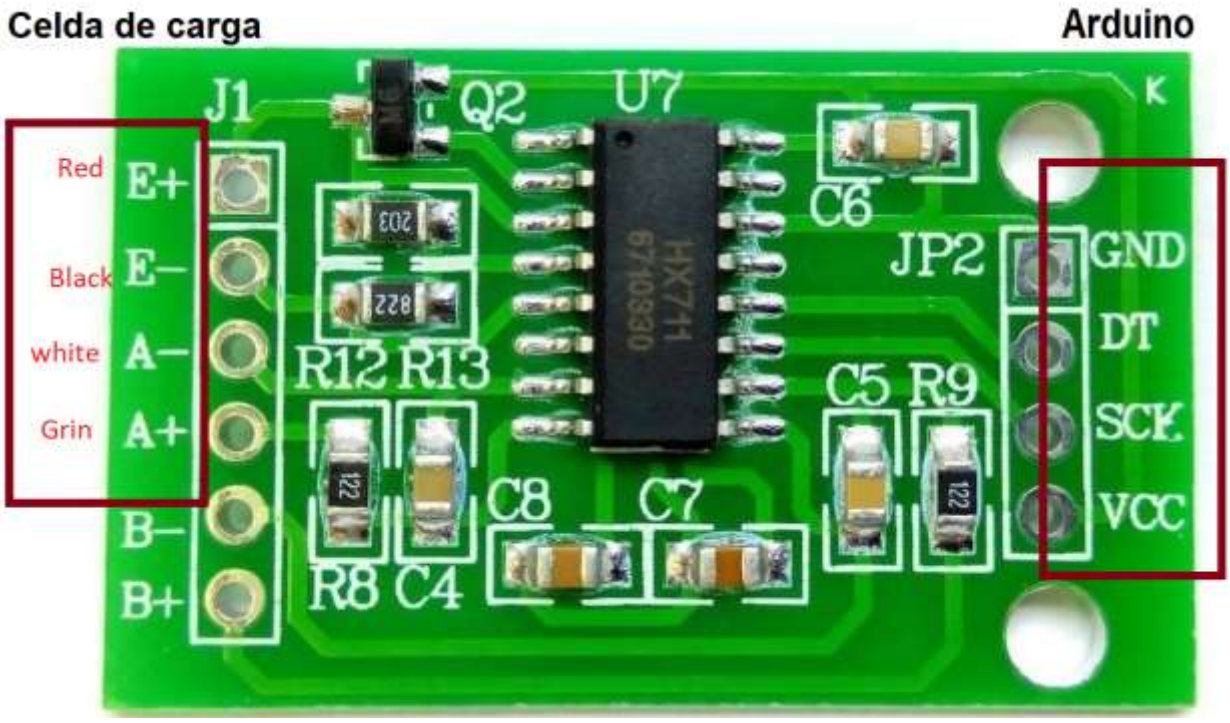
[80] Hernández Márquez, Felipe, Benjamín, Rodríguez Medina y Alfonso Torres Ríos. 12BC. "Puente de Wheatstone Modificado Utilizado En Una Comparación Internacional En El Valor de 1 GΩ". {En línea}. {24 octubre 2020} disponible en: http://cenam.mx/simposio2008/sm_2008/memorias/M2/SM2008-M233-1178.pdf.

carga corresponden a cada una de las resistencias presentes en este esquema, pero para poder lograr el acople entre el la etapa de captación de la variable física del peso del usuario, con el microcontrolador Arduino, es necesario conectar el puente Wheatstone con un amplificador. ya que normalmente para las celdas de carga de precisión estándar, como es el caso de este trabajo de grado, se utiliza el módulo HX711, Este módulo tiene esta característica y permite una conexión sencilla ocupando tan solo 2 pines del microprocesador, estos se explican un poco más adelante en el documento.

Como se mencionó en el párrafo anterior el módulo HX711 es el encargado de obtener recibir los valores de voltaje, amplificarlos y transmitirlo al Arduino. Esto se hace con la finalidad de que dentro del Arduino se pueda realizar la operación de proporcionalidad y conocer el peso que está soportando la báscula en su parte superior.

El módulo Hx711 está compuesto por dos hileras de pines. Una sirve para conectar al Arduino mientras que los otros pines sirven para conectar las celdas de carga como se observa en la imagen 56.

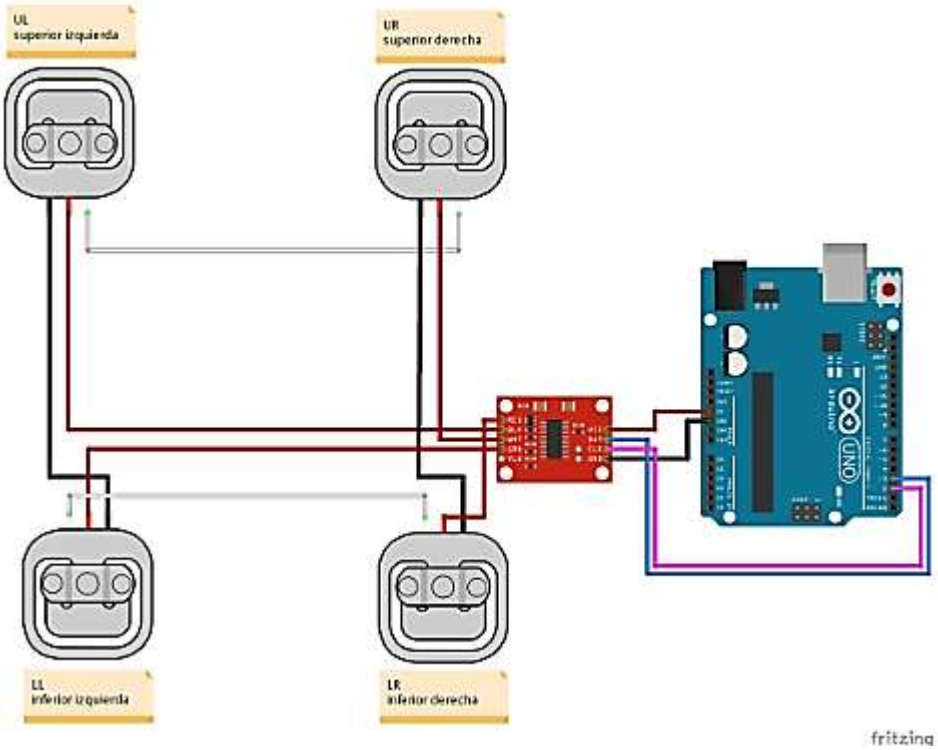
Imagen 56. Módulo amplificador HX711, y las hileras de pines.



.
Fuente: https://programarfacil.com/blog/arduino-blog/hx711-arduino-bascula-digital/

El proceso entonces inicia con la conexión de las celdas de carga con el módulo amplificador HX711, se muestra en la Imagen 57.

Imagen 57. Esquema de conexión completo para funcionamiento de la báscula.



Fuente: https://www.hbm.com/es/7163/el-puente-de-wheatstone-galgas-extensometricas/.

Para este Trabajo de grado se van a utilizar 4 celdas de carga cada una tiene 3 cables (cable negro(tierra), cable blanco (polaridad positiva) y cable verde(salida) que sustituye a el cable rojo en el esquema anterior). Los cables se conectan como en la Imagen 57 ya que permite conectar las 4 celdas para conectarlas al módulo amplificador HX711, emulando el puente Wheatstone.

La posición de cada una de las celdas de carga se determina por la posición en el esquema de la imagen anterior proporcionado por Luis del Valle Hernández. Para lo que es necesario definir que es superior (Upper), inferior (Lower), izquierda (Left) y derecha (Right). Lo anterior se hace para que la ubicación y conexión correcta con respecto a la Imagen 57 en donde es necesario identificar las posiciones Superior Izquierda (UI), superior derecha (UR), inferior izquierda (LL) e inferior derecha (LD).

Para este Trabajo de grado la parte superior es como se muestra en la Imagen 56. lo que permite organizar y conectar los cables como se mostró en la Imagen 57 al

módulo HX711. Una vez los sensores están conectados al módulo. Se realiza la conexión del módulo amplificador al microcontrolador Arduino Uno como se observa en la tabla 6

Tabla 6. Pines de conexión Arduino con amplificador HX711.

Pines Arduino	Pines Módulo HX711
Pin 2 Digital (Salida Entrada de los datos)	Pin Data
Pin 3 Digital (Salida Reloj)	Pin Clock
Pin GND	Pin GND
Pin 5 V	Pin Vcc

Fuente: Autor.

Una vez hecho lo anterior, se estructura la lógica de programación que permita calibrar y obtener el peso.

Para poder iniciar se debe instalar una librería capaz de manejar o gestionar el módulo HX711. Para poder instalarla es necesario ir al menú “Programa” luego en “Incluir librería”, después “Administrar librería”. Dentro del Administrador de librerías de Arduino se hace la búsqueda “HX711” y se instala la opción de la librería hecha por Bogdan Necula y Andrea Moti.

Una vez terminado el proceso de instalación de las librerías necesarias, se procede a la codificación en el IDE de Arduino para recibir la salida del módulo HX711 que se encarga de entregar el valor del peso percibido por las celdas de carga. Esto es posible al asignar dos pines digitales de la placa Arduino Uno para obtener la salida del módulo amplificador HX711 (Pin 2) y otro pin para generar una señal de reloj que permita la comunicación entre el módulo y la placa Arduino (Pin 3).

6.5.1 PROCESO DE CALIBRACIÓN DE LA BÁSCULA.

1. Quitar peso de la báscula antes de cargar el código.

2. Una vez que se empiezan a enviar los datos y se visualiza, se pone algo sobre la báscula con un peso conocido (Se utilizó 1 Kg de arroz).

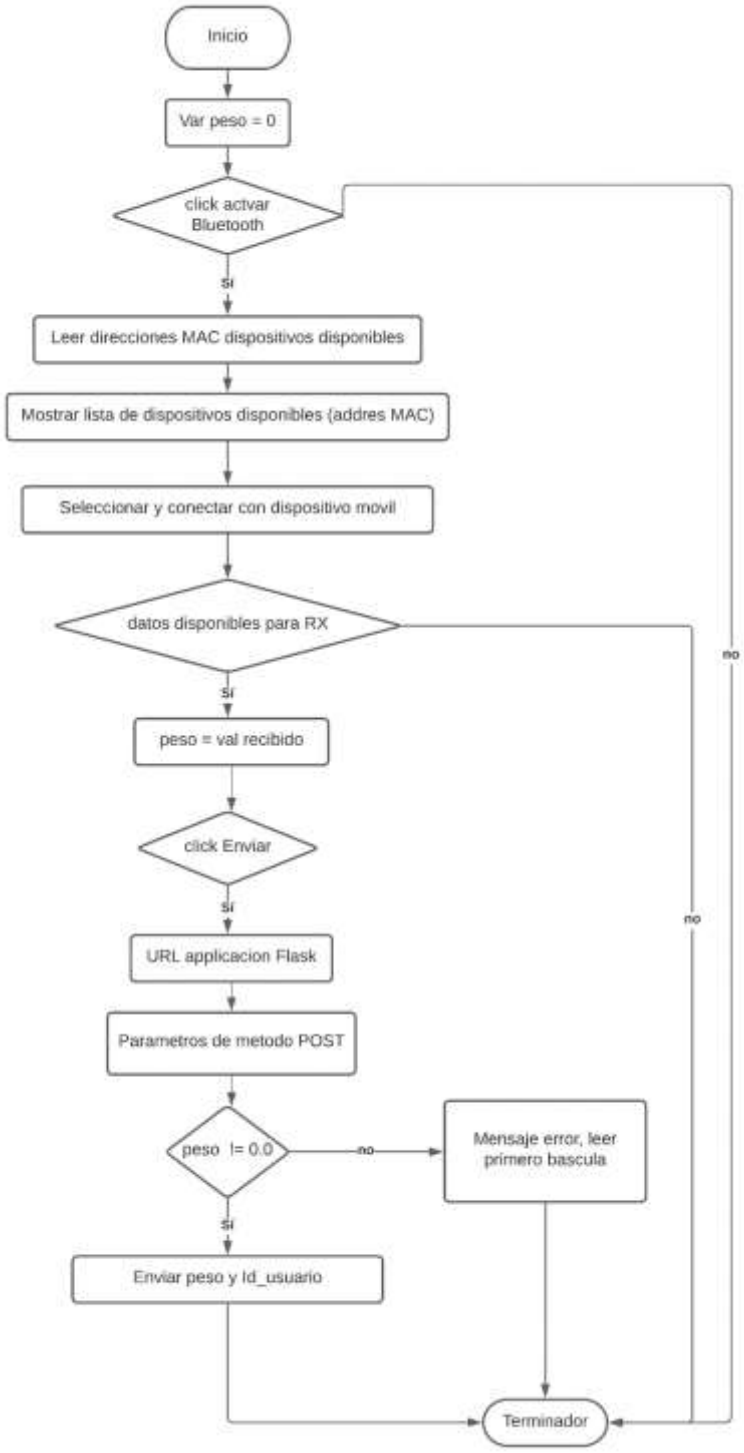
3. Se ajusta el factor de calibración escribiendo más (+) o menos (-) en el monitor serie.

Es necesario identificar el valor de calibración propio para ajustar las mediciones y este factor de calibración es propio de cada báscula. Para hacer la calibración de la báscula, el valor obtenido por el módulo HX711 coincida con el valor del peso de referencia (1 Kg), se almacena ese valor del factor de calibración, para este trabajo de grado el factor fue 25880.

Una vez concluido este proceso, se procede a la transmisión de los valores obtenidos por la báscula desde un módulo Bluetooth (Hc-06) a un dispositivo móvil con sistema operativo Android que se encargará de subir los valores a la aplicación desarrollada con Flask para almacenar su Valor de peso en la base de datos.

El celular que se encargará de cumplir con esta funcionalidad debe contar con una aplicación generada en la página de App Inventor desarrollada por la MIT dentro de App inventor. A continuación, se explica la construcción de la aplicación hecha en MIT APP Inventor 2 como se muestra en la Imagen 59 teniendo en cuenta el diseño que se observa en la imagen 58.

Imagen 58. Diseño funcional Bascula.



Fuente: Autor

En el diagrama que se muestra en la imagen 59, se observa paso a paso el diseño para la recepción y transmisión de los datos a la aplicación de Flask, recibidos por la báscula a través del protocolo Bluetooth (802.3), para posteriormente la transmisión de la información al servidor por medio del protocolo (802.11) o por una red móvil como puede ser 3G o 4G.

El diseño que se implementa en esta etapa, ya teniendo, funcionando la báscula calibrada para poder recibir los valores de peso. Se inicia el proceso creando una variable global en app inventor que se encargara de recibir los valores del peso, posteriormente se pregunta si el usuario ha realizado la acción de oprimir el botón de activar, en caso tal de ser cierto, se despliega la lista que ha leído con anterioridad el dispositivo con la dirección MAC y sus respectivos nombres de equipos.

Una vez el usuario ha seleccionado la dirección Mac a la que desea acceder, se realiza el proceso de sincronización entre el dispositivo móvil y el módulo HC-06 que permitirá la conexión con la báscula.

Cuando ya se genera una conexión entre el dispositivo móvil y la báscula, se verifica si hay datos para recibir en el buffer de recepción; en caso de que sea un sí, se asigna el valor en peso y se muestra en pantalla, de lo contrario el valor por default es 0.0.

Si el usuario presiona el botón de enviar la información viajará a través del protocolo TCP/IP específicamente en la capa de aplicación, debido a que se hará él envió de información en formato Json u otros formatos. Por lo que se prepara el URL que se va a utilizar para acceder a la aplicación alojada en la nube y se le dice que tipo de cabecera HTTP, va a tener. Antes de realizar el envío de valor almacenado en la variable peso, se valida que este sea diferente a 0.0, por lo que se asume que ya hay valores disponibles para la transmisión.

Imagen 59. Boceto visualización de la pantalla del dispositivo móvil.

Fuente: Autor.

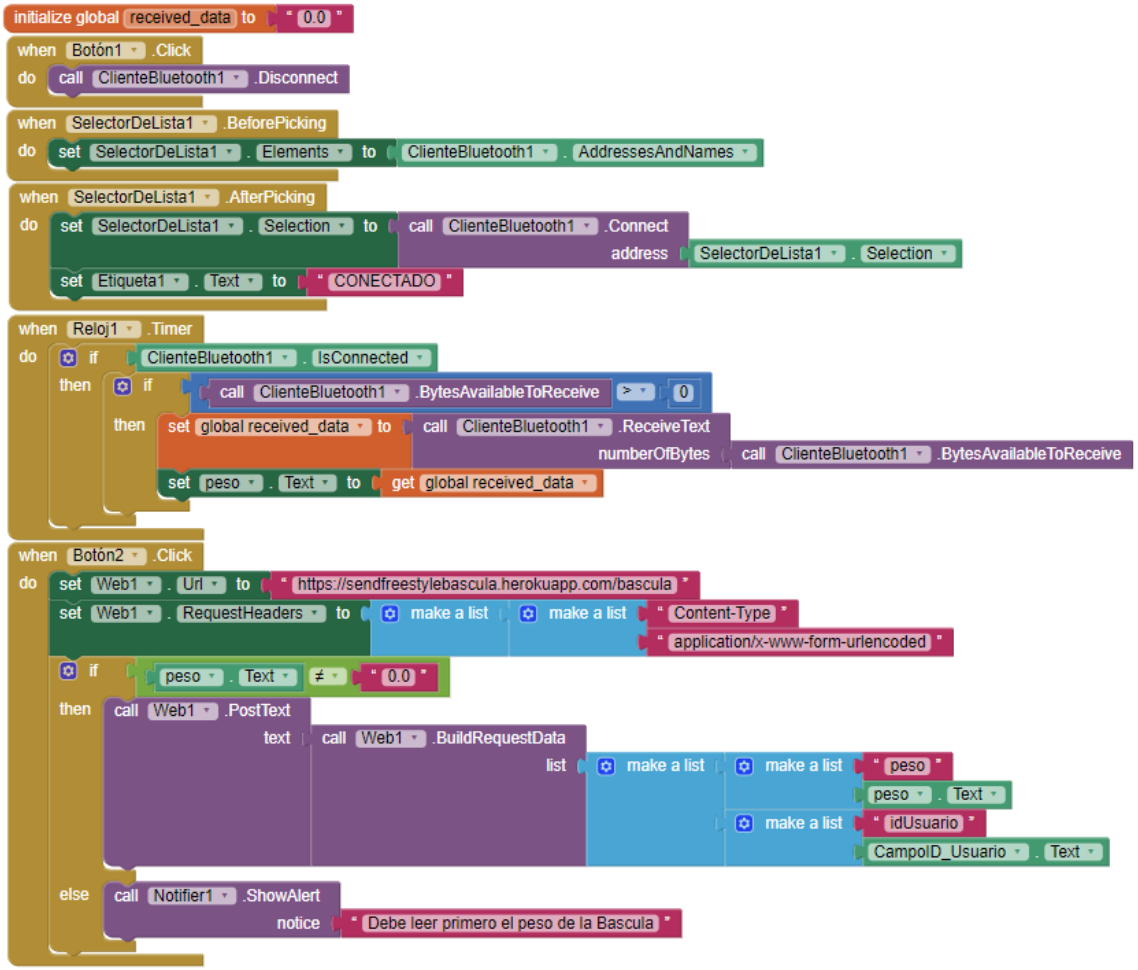
La imagen 59 muestra los componentes que se han agregado, estos componentes son:

- Pantalla 1
  - Disposiciòn Horizontal 1
    - Selector De lista1
    - Boton1
  - Horizontal arrangement 3
    - Label 4 (“Ingrese su número de identificacion:”)
    - CampoID_Usuario(Campo de texto)
  - Horizontal arrangement 1
    - List Picker (Con logo de Bluetooth)
  - Horizontal arrangement 2
    - Label_Masa_usuario(“Masa usuario:”)
    - Label Peso(“Peso”)
    - Label3(“Kg”)
  - Disposiciòn Horizontal 2
    - Botòn 2 (“Enviar”)
    - Etiqueta 1(Para confirmar conexión Bluetooth con texto “Conectado”)
    - Reloj 1.

- Cliente Bluetooth 1
- Notifier 1
- Web 1

Al ingresar todos estos componentes mencionados anteriormente en la pantalla 1. Es la única que se va a utilizar para el caso de la aplicación que se usará en este Trabajo de Grado. Se procede a la programación en bloques de la aplicación que se muestra en la Imagen 60.

Imagen 60. Programación en Bloques MIT App Inventor 2.



Fuente: Autor

En la lógica implementada en los bloques de App Inventor 2 inicia con una variable Global Inicializada Received_Data con el valor de 0.0, se inicializa con este valor para que el usuario de la aplicación entienda que aún no se ha recibido ninguna variación en el peso sensado por la báscula. Una vez se reciba un valor de la báscula este se actualizará en el Label “Peso” en la pantalla gracias al bloque “When (Reloj1). Timer”.

Bloque **When (Botón1). Click**:

En este bloque se tiene el comando para desconectar las conexiones a Bluetooth existentes y se ejecuta cuando es oprimido (se da click) en el botón 1(“Apagar”).

Bloque When (SelectorDeLista1). BeforePicking:

En este bloque se encarga de obtener y preparar para listar todas direcciones capa física de las conexiones Bluetooth disponibles.

Bloque When (SelectorDeLista1). AfterPicking:

En este bloque se despliega en una pantalla negra y letras blancas una lista de los dispositivos y su respectiva dirección física, con los que puede establecer una conexión Bluetooth, de estas opciones se selecciona para el caso de este laboratorio la opción que contiene en su nombre HC-06 (Hay que tener en cuenta que para que suceda esto debe estar encendido el Bluetooth del dispositivo móvil).

Bloque When (Reloj1). Timer:

Este bloque es el encargado de la recepción de los datos que son enviados de la báscula por medio del módulo HC-06. Validando primero si el cliente Bluetooth del celular está conectado y si lo está, revisa si se ha hecho alguna llamada para recibir datos (Superior a 0 bits), si es así se asigna los datos a la variable global “receuved_data” y después este valor se asigna al Label “Peso” para ser mostrado en pantalla para la visualización del usuario.

Bloque When (Botón2). Click:

En este se utiliza el componente “WEB 1” para establecer una url para acceder a la aplicación desarrollada con el mini Framework Flask en este trabajo de grado y entregarle la información del peso y el valor de documento de identidad, también se hace un Request Headers en el que se le envía una lista con los parámetros “Content-Type” que es el tipo de conexión y “application/x-www-form-urlencoded” necesarios para conectar la aplicación con el servidor.

Después va una instrucción IF en la que se valida si el valor de peso es diferente de 0.0 con el fin de revisar que haya una lectura previa a la acción de envío de peso al servidor. dentro del If cumpliéndose la condición anterior, se procede a realizar el envío de los datos por medio del método POST de un HTML, estando los datos dentro de una lista compuesta por una pareja de nombre de la variable que identificará el servidor que recibe la acción POST y su respectivo valor, y el conjunto de estas parejas de nombre de la variable y valor, estará dentro de otra lista que ejecuta la llamada al método POST. En caso de que la condición del IF no se cumpla

se mostrará un alert, encargado de dar el mensaje de que “debe leer primero el peso de la báscula” antes de enviar los datos.

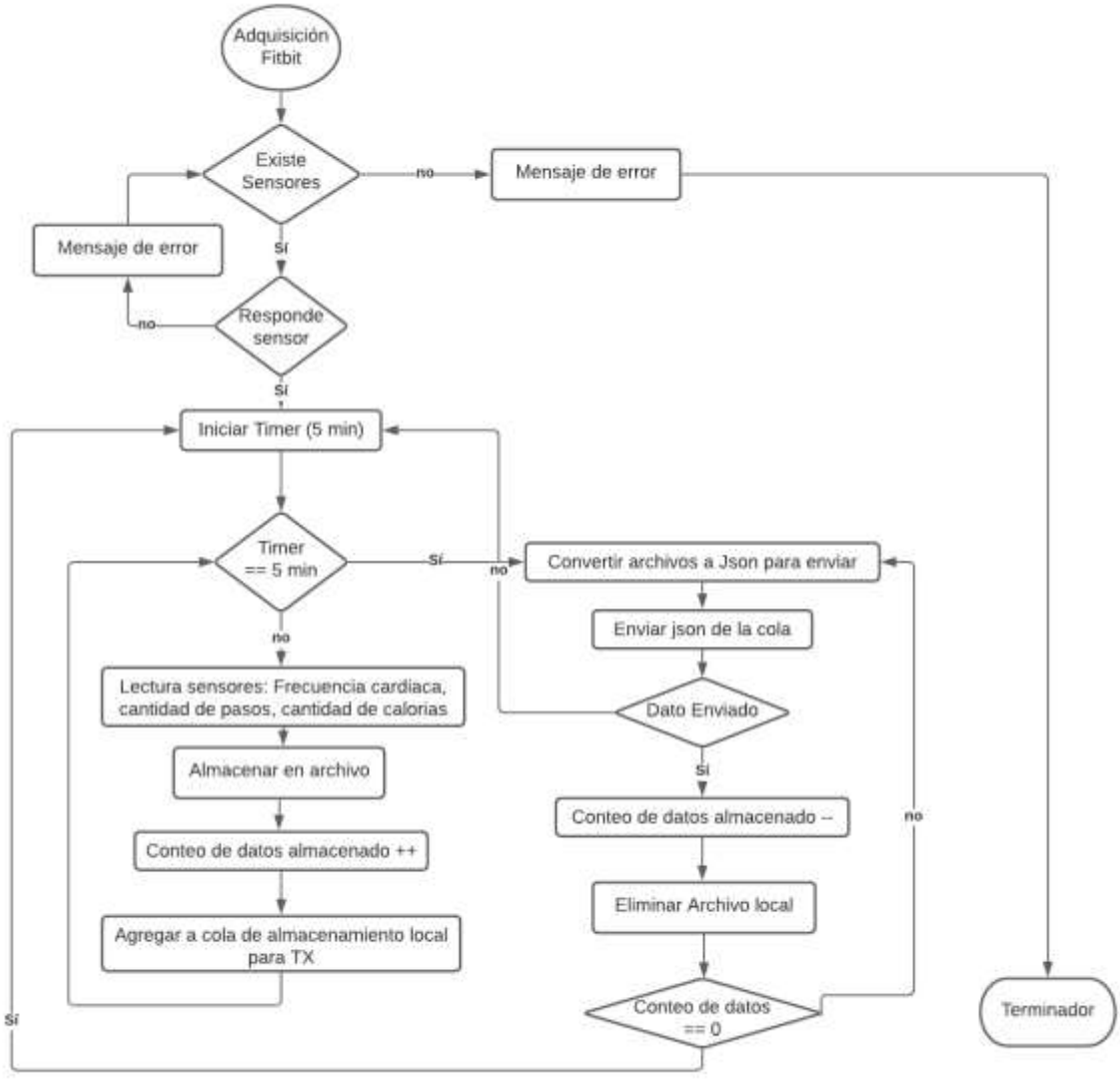
6.6 DESARROLLO FITBIT PARA EXTRACCIÓN DE VARIABLES (CANTIDAD DE PASOS, TASA DE CALORÍAS, FRECUENCIA CARDIACA)

6.6.1 DISEÑO FITBIT

Lo que se observa en la imagen 61, es el esquema completo que se implementa en el Fitbit, el diseño se centra en poder acceder a los valores de los sensores del Fitbit y poder extraerlo para poder enviarlo a la aplicación alojada en Heroku, esto se realiza en intervalo de tiempos de 5 min, donde se valida que cada uno de los datos se envíe, si es el caso de que se envió correctamente, se resta el conteo de datos almacenados localmente al igual que el archivo local.

De lo contrario los archivos siguen en cola, se reinicia el temporizador para volver a realizar un nuevo intento de conexión y enviar toda la información.

Imagen 61. Diagrama funcionamiento Fitbit.



Fuente: Autor.

Se comienza la explicación del proceso realizado con el dispositivo Fitbit y su respectivo SDK. La documentación que ofrece Fitbit para los desarrolladores, en donde ofrece un SDK donde se puede crear aplicaciones y carátulas de reloj para dispositivos con OS Fitbit, como Fitbit Ionic, Versa, Versa Lite y Versa 2 [81].

Este SDK permite gestionar las características del dispositivo Fitbit, lo que permitirá extraer los valores de las variables fisiológicas que puede obtener el Fitbit ionic.

[81] Guía de arquitectura de aplicaciones Fitbit. {En línea}. {25 octubre de 2020} disponible en: https://dev.fitbit.com/getting-started/

Es necesario:

Se necesitará los siguientes requisitos previos para desarrollar o carátulas de reloj con el SDK de Fitbit:

- Cuenta de usuario de Fitbit. la cual se puede crear directamente en la página de Fitbit.
- Un dispositivo Fitbit OS o el Fitbit OS Simulador para Windows o macOS.
- La última aplicación móvil de Fitbit para Android, iOS o Windows Phone. emparejada con el dispositivo Fitbit el cual puede ser suplantado por el simulador para realizar las pruebas
- Una computadora con acceso a Fitbit Studio.
- Una red inalámbrica para proporcionar al dispositivo Fitbit una conexión a Internet.

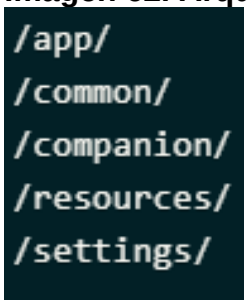
Antes de iniciar:

Para iniciar es necesario conocer la arquitectura que ofrece Fitbit y a su vez cómo se relacionan sus distintos elementos y sensores para poder llevar a cabo el diagrama que se encuentra en la imagen 61.

6.6.2 ARQUITECTURA SDK FITBIT CONSTA DE UNA ESTRUCTURA EN CARPETAS:

Esta estructura en carpetas (ver imagen 62) hace posible que en el proceso de compilación agregue correctamente la aplicación para su instalación en el dispositivo. A su vez compila el complemento y las respectivas configuraciones para la instalación en el dispositivo móvil de ser necesario.

Imagen 62. Arquitectura SDK Fitbit.



Fuente: https://dev.fitbit.com/build/guides/application/

6.6.2.1 Tamaño y limitaciones

Es relevante saber que el tamaño máximo de una aplicación en el momento de su instalación corresponde a 10 megabytes. Así como también es importante saber

que el tamaño total ocupado del sistema de archivos utilizado por una instalación está limitado a 15 megabytes, esto incluye los recursos y también los archivos escritos con la API perteneciente al sistema de archivos

6.6.2.2 Lenguaje de desarrollo

Para las carpetas */app/* - */common/* - */companion/* pueden contener diversos archivos tanto en JavaScript (extensión .js) o TypeScript (extensión .ts).

Mientras se realiza el proceso de compilación, los scripts son automáticamente compilados, empaquetados y optimizados mediante el compilador de Typescript y rollus.js. Esto genera un único archivo ECMAScript 5.1 para la aplicación y otro archivo para el companion. Para la ejecución de JavaScript se hace mediante el **motor JerryScript[82].**

Para la carpeta */settings/* solo debe contener un archivo **React JSX**, con el nombre de index.jsx.

6.6.3 CARPETAS DEL PROYECTO FITBIT

6.6.3.1 Aplicación (/app/)

La carpeta */app/* contiene lo que concierne a la lógica de la aplicación que corre en el dispositivo. Una característica importante es que el código que contiene esta carpeta tiene acceso a la **API** del **Dispositivo (Todas las características de hardware)**, contando con la propiedad de poder interactuar directamente con la capa de presentación del dispositivo, al igual que comunicarse con el companion o leer y escribir la carpeta de configuraciones (settings).

6.6.3.2 Companion(/companion/)

La carpeta /companion/ es la encargada de contener la lógica complementaria que se ejecuta en el dispositivo móvil. El código alojado en esta carpeta tiene acceso a la **API companion** y es capaz de realizar solicitudes directamente a Internet y comunicarse con la aplicación.

6.6.3.3 Código compartido(/common/)

La carpeta /common/ se pueden repartir entre la aplicación y el companion para minimizar la duplicidad de código, esto con el objetivo de reducir el tamaño, teniendo en mente el tamaño límite de desarrollo. crear cada archivo como un módulo ES6 (JavaScript), luego es necesario importar el módulo a su aplicación o companion. En el proceso de compilación, se realizará automáticamente el tree shaking para

[82] Guía de arquitectura de aplicaciones Fitbit. Óp. cit., p. 1

excluir cualquier módulo que no se está utilizando en el resultado final, esto es otra herramienta que permite optimizar el código compilado.

6.6.3.4 Recursos (/recursos/)

La carpeta contiene todos los recursos que se incluirán con la aplicación durante el proceso de la compilación.

6.6.3.5 Configuraciones(/settings/)

La carpeta contiene las características escritas necesarias para la configuración de la aplicación, estas son escritas usando React JSX. Esto tiene como objetivo que los usuarios puedan configurar una aplicación. La codificación dentro de esta carpeta y/o archivo, tiene acceso a la API Settings**.**

Una vez explicada la arquitectura del SDK para el Fitbit, lo siguiente es abordar el funcionamiento de la solución generada en el SDK para transmitir los datos del dispositivo Fitbit hacia el servidor MySql, utilizando un webSocket. El dispositivo Fitbit necesita estar conectado a la red wifi para poder enviar los datos por medio del webSocket, en caso tal de no tener acceso a la red, el dispositivo de configuro para que almacene los datos hasta un próximo intento de conexión que está definido por un time out dentro de la carpeta app, y en caso tal de que no se conecte, continúa almacenando los datos hasta que se tenga nuevamente una conexión.

Para el almacenamiento correcto de las variables fisiológicas lo primero que se hace es analizar si el usuario está utilizando el dispositivo o no, ya que, si no lo está utilizando, no tiene sentido almacenar datos nulos que gasten espacio en memoria y generen basura en la base de datos.

También se encuentra la función countdown() que se encarga de llevar el conteo del tiempo entre cada intento de conexión con la red para enviar los datos almacenados, esta función tiene la funcionalidad de timer o time out, llevando el conteo de la ventana de tiempo seleccionada de 5 minutos. Esta función estará encargada de hacer el llamado a la función takeData() que se encarga de leer los valores sensados de las variables fisiológicas (tasa de calorías, cantidad de pasos, frecuencia cardiaca).

Una vez obtenidos los valores de las variables fisiológicas, se procede a hacer el llamado de la función save() destinada para almacenar los datos de forma local en el Fitbit. El almacenamiento de los datos se hace mediante archivos tipo Json, debido a que es un formato estandarizado y a su vez fácil de manejar para varios lenguajes al ser una estructura clave-valor, ademas el espacio ocupado por este formato es mínimo lo que permite que sea viable el almacenamiento de los registros en la memoria del Fitbit cuando se encuentra offline, para su posterior transmisión al tener conexión nuevamente. Dentro de esta función también es importante

mencionar que se lleva a cabo un conteo de los archivos almacenados y así saber con exactitud la cantidad de datos a enviar.

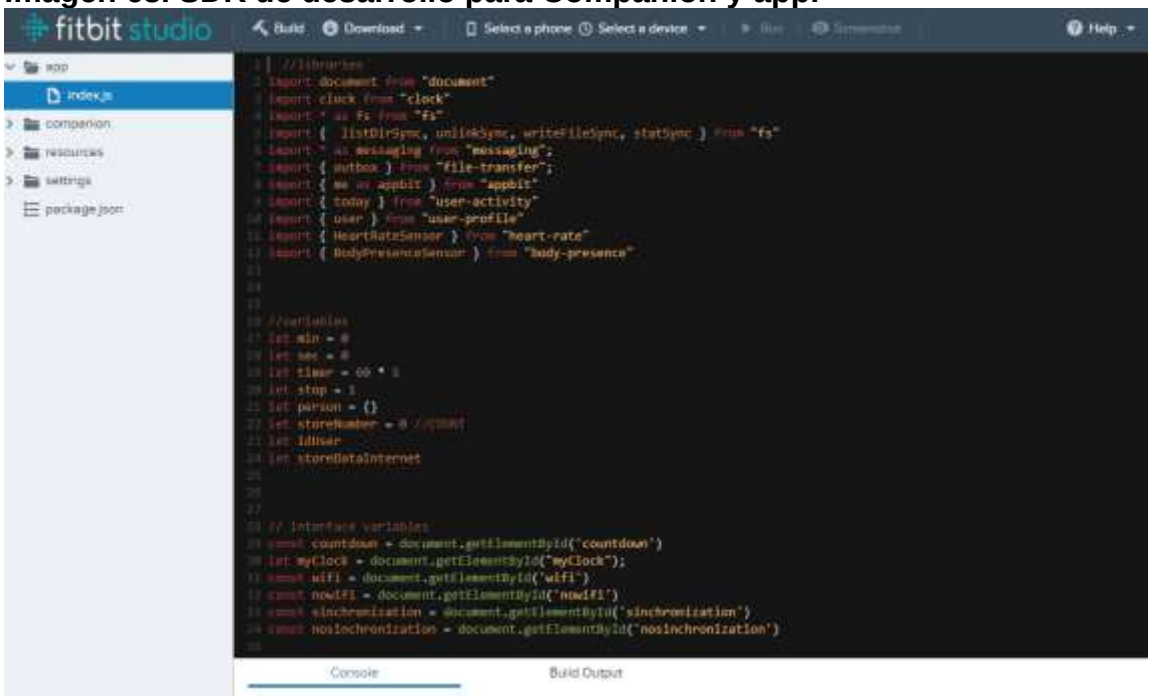
Dentro del archivo app, la función connection() encargada de la conexión a internet y la transmisión de los datos, en donde primero se realiza la acción de conexión de forma sincronizada con la red y el companion (que se encuentra el dispositivo móvil), una vez se comprueba que la conexión existe se procede al envío de la información almacenada con la función sendDataStore().

La función sendDataStore() pregunta si la bandera que indica si hay archivos almacenados está en 1, si es el caso procede a realizar la acción de acceder a los datos almacenados de forma local y carga los archivos json para enviarlos al webSocket que se encuentra dentro del companion. Esto se hace por motivos de seguridad, debido a que los dispositivos IoT en la actualidad no cuentan con seguridad informática, es por esto que se utiliza el dispositivo móvil para poder brindar la seguridad necesaria del envío de la información hasta el servidor por medio de un webSocket, y una vez se hace él envió del archivo que contiene los datos, se hace el llamado a la función deleteDataStore().

La función deleteDataStore() se encarga de que al enviar un dato al webSocket, este dato sea eliminado de los archivos locales, para evitar posibles errores por redundancia en el almacenamiento de los datos.

Todo lo anterior se realiza dentro del SDK como se observa en la imagen 61. Donde se encuentra la estructura de la aplicación que después se instala en el Fitbit y el dispositivo móvil para su correcto funcionamiento.

Imagen 63. SDK de desarrollo para Companion y app.



Fuente: Autor.

7. VALIDACIÓN

En esta sección se lleva a cabo las pruebas que permiten validar el correcto funcionamiento del diseño que permite el monitoreo remoto, también centralizando el conjunto de variables en una sola base de datos**.** Primero se validará la calibración de la báscula colocando sobre ella valores de referencia (0.5 Kg, 1 Kg) además de la medición de peso de una persona de aproximadamente 1.70 m (76 Kg), validándose en paralelo la conexión entre la báscula y el dispositivo móvil por medio de la tecnología Bluetooth. Para la visualización del peso se hace por medio de la pantalla del celular sincronizado a la báscula, una vez leído el valor de peso se compara con los valores de referencia de peso y se realiza la acción de envió hacia el servidor donde está desplegada la aplicación web.

Se realiza una segunda validación de para verificar la integridad de los datos desde que son enviados al servidor, verificando que llegue correctamente el valor del peso, y después que sea almacenado correctamente en el servidor remoto de MySQL prestado por ClearDB. Un proceso similar se realiza para la validación del sensor Freestyle Libre.

La validación consta de la comparación entre los valores almacenados en la base de datos MongoDB que es la encargada de recibir los datos enviados desde la aplicación Glimp. En la aplicación web se valida que haya leído la misma cantidad disponible en la colección de MongoDB, una vez la aplicación hace él envió a el servidor MySQL se revisa que los valores estén siendo almacenados correctamente contrastando los datos en ambas bases de datos. La comprobación o lectura de la base de datos MySQL, se realiza con el software WorkBench.

7.1 VALIDACIÓN BÁSCULA.

Como se mencionó anteriormente, en la báscula se realizó la validación de los datos sensados, su visualización se obtiene por medio de la aplicación desarrollada en app inventor 2, como se puede observar en la imagen 65. La validación inicia colocando un peso de referencia para esta primera parte será una bolsa de harina de trigo con un peso de 0.5 Kg (ver imagen 64), este se coloca unos segundos después de que se inició la báscula y se calibro automáticamente, al colocar la libra de harina sobre la báscula, transmite el valor del peso a través del módulo Hc-06 al celular, el cual lo recibe y muestra, comprobando que si hay conexión y es exitosa entre el Arduino uno y el dispositivo móvil.

Imagen 64. Bascula con 0.5 Kg de peso.

Fuente: Autor.

Como se observa en la imagen 65, el valor recibido corresponde a 0.5 Kg lo que corresponde al valor correcto del medio kilo de harina colocado sobre la báscula. Continuando el proceso sigue la validación de la integridad de la información desde el dispositivo hasta el servidor Heroku que aloja la aplicación Web encargada del almacenamiento.

Imagen 65. Recepción de peso 0.5 Kg de la bascula

Fuente: Autor

En la imagen 66, se observa la acción de envió del valor de peso al servidor y el aviso que confirma él envió.

Imagen 66. Envió del peso a la aplicación web (servidor)

Fuente: Autor

Una vez que se realiza él envió, como se observa en la imagen 65. Se realiza la verificación del valor recibido por el dispositivo móvil en la cola de registros del servidor Heroku donde está la aplicación web en ejecución, esto se observa por medio del símbolo del sistema (CMD). En la imagen 65, con un rectángulo rojo

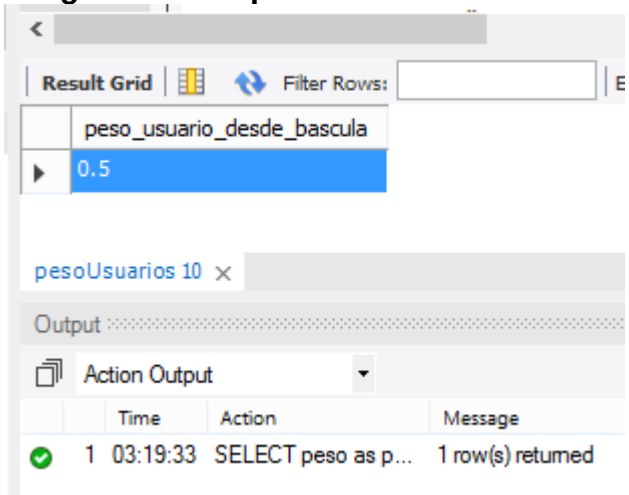
pequeño el valor que se recibió y que se preparó para la sentencia SQL para la inserción en la base de datos. La inserción de la base de datos se corrobora en la Imagen 67, al mostrar el campo de la tabla “pesoUsuarios” donde se ve los datos almacenados, en este caso solo hay uno para ir mostrando paso a paso la inserción de múltiples valores y su integridad en el almacenamiento.

Imagen 67. Registro de acciones y peticiones de la aplicación en el servidor.

Fuente: Autor.

Una vez se valida en la imagen 67, el valor enviado por la pesa (ver Imagen 64) en servidor por medio de un api rest genera una conexión a la base de datos e inserta el valor de la base de datos en la tabla “pesoUsuarios”, el valor almacenado en la base de datos MySQL (ver imagen 68), es coherente con el enviado en la imagen 65, recibido y reenviado en la imagen 67.

Imagen 68. Comprobación del almacenamiento del valor correcto en MySQL



Fuente: Autor

Se procede a cambiar el peso sobre la báscula aumentando el peso de referencia a 1 Kg colocando harina de maíz sobre la báscula (observar imagen 69) y obtiene el valor del peso en el dispositivo móvil a través de Bluetooth.

Imagen 69. Bascula con 1.0 Kg de peso.

Fuente: Autor

El valor medido en la báscula se encuentra en Imagen 69 de 1 Kg. Se envía nuevamente al dispositivo móvil como se observa en la Imagen 70, donde se visualiza una correcta lectura del valor del peso obteniendo como resultado 1.0 Kg en el espacio reservado en la pantalla para visualizar este Valor.

Imagen 70. Recepción de peso 1.0 Kg de la bascula

Fuente: Autor

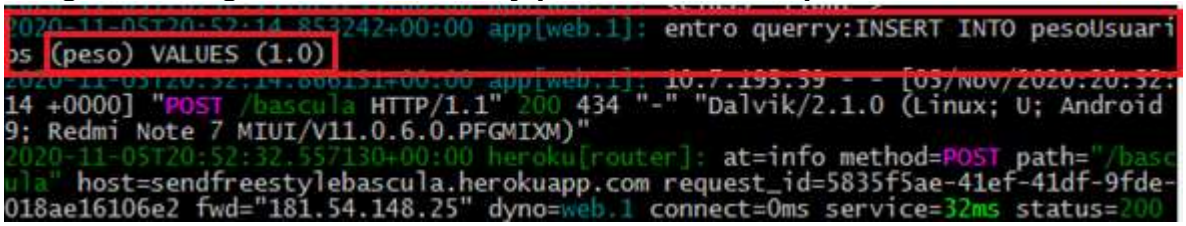
Una vez se ha obtenido la lectura de la báscula se realiza el proceso de envió el cual se observa en la imagen 71, donde se muestra un mensaje que confirma él envió del peso al servidor una vez se oprime el botón de “ENVIAR”.

Imagen 71. Envió del peso a la aplicación web (servidor).

Fuente: Autor

Al enviar los datos de la aplicación de app Inventor al servidor, se revisa la cola de registros del servidor específicamente de la aplicación, como se observa el resaltado en la Imagen 72, el valor recibido es 1.0, y se ejecuta la acción de inserción a la base de datos con una sentencia SQL como se puede ver en la misma imagen.

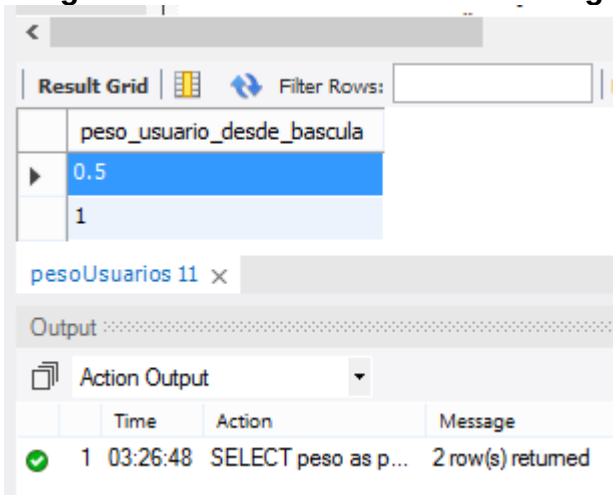
Imagen 72. Registro de acciones y peticiones de la aplicación en el servidor.



Fuente: Autor

Continuado con la validación, se verifica que el valor sea insertado correctamente en la base de datos por lo que nuevamente se revisa la tabla Usuario en el campo destinado para el peso, donde se revisa que este almacenado el valor y que, a su vez, este corresponda con el valor correcto enviado por la báscula. El valor que se registra en la base de datos de acuerdo con la imagen 73, es 1 Kg y es coherente con el valor enviado desde la báscula.

Imagen 73. Valor almacenado de 0.5 Kg en base de datos.



Fuente: Autor

La próxima validación corresponde al envió de la información del peso de una persona de prueba con una estatura de aproximadamente 1.70 m, que se ubica exactamente cómo se puede observar en la imagen 74. Para el sensado de su peso.

Imagen 74. Medición de peso de persona.

Fuente: Autor.

El valor de peso sensado de la persona de prueba por la báscula, corresponde a 76.4 Kg, valor que se visualiza en la imagen 75. Por lo que se continua con el proceso y se envía la información al servidor para su almacenamiento (ver imagen 76).

Imagen 75. Recepción de peso 76.4 Kg de la báscula.

Fuente: Autor.

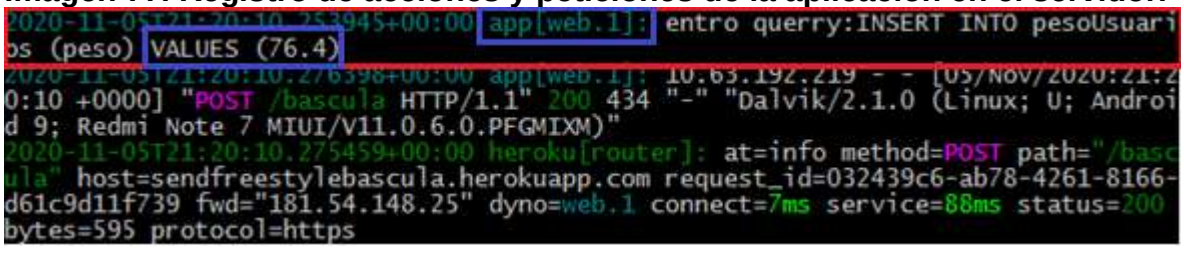
El valor sensado de 76.4 Kg se envía al servidor utilizando el botón de “ENVIAR” lo que arroja un aviso de que se pudo enviar satisfactoriamente los datos al servidor. Sé que procede nuevamente a la revisión del símbolo del sistema en donde se encuentra el registro de actividad de la aplicación implementada en el servido Heroku (revisar Imagen 76).

Imagen 76. Envió del peso a la aplicación web (servidor).

Fuente: Autor.

Al hacer él envió del dispositivo móvil, se valida que el valor llegue correctamente al servidor, esto por medio de la cola de registros que brinda el servidor Heroku, donde se observa en la Imagen 77, que el valor de 76.4 llega correctamente y se realiza la acción de inserción en la base de datos para guardar el registro del peso de la persona.

Imagen 77. Registro de acciones y peticiones de la aplicación en el servidor.

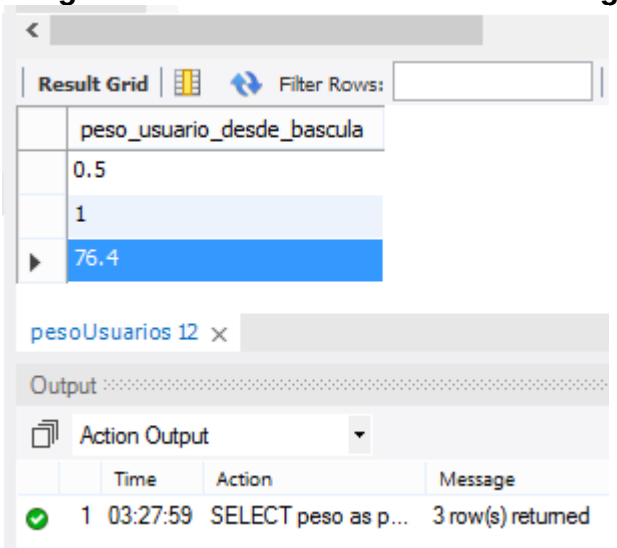


Fuente: Autor

En la Imagen 78, se observa el valor almacenado en la base de datos. El valor registrado en la tabla de pesoUsuarios, es coherente con la etapa de obtención de

los datos realizada por la báscula, siendo enviado correctamente desde el dispositivo móvil hacia el servidor y este realizando la acción de su almacenamiento correcto.

Imagen 78. Valor almacenado de 76.4 Kg en base de datos.



Fuente: Autor.

Tabla 7. Tiempo de datos enviado al servidor

Peso (Kg) enviado	Tiempo en llegar al servidor y ser leído.
0.5	32 ms
1	32 ms
76.4	85 ms
1	82 ms
72.9	110 ms
76.6	69 ms
71.3	66 ms

Fuente: Autor.

En la tabla 7 se observa que el tiempo de envió, aunque presenta retardos aleatorios, no superan el segundo de transmisión.

7.2 VALIDACIÓN DATOS ALMACENADOS DE FREESTYLE LIBRE.

Para realizar la validación de los datos obtenidos por el sensor FreeStyle, también transmisión para su almacenamiento y traspaso de los datos a una base de datos MySql, Realizando una comparación entre los valores almacenados en la base de datos MongoDB y la base de datos MySQL donde se almacenan todos los datos de las variables fisiológicas.

Para iniciar la validación de la transmisión y almacenamiento del sensor Freestyle libre, se observa la aplicación Glimp después de leer el sensor Freestyle libre, en esta lectura el sensor le entrega a el celular el valor de 124 mg/dl de glucosa, esto lo hace por medio de la tecnología NFC, para este trabajo se utilizó un Xiaomi mi 9 T Pro que cuenta el con esta tecnología integrada. Una vez se leyó el valor con la aplicación Glimp (ver imagen 79) y está conectada con la aplicación brindada de la comunidad Nightscout implementada en el servidor Heroku, para hacer seguimiento remoto.

Imagen 79. Lectura sensor Freestyle libre Abbott con Glimp.

Fuente: Autor

Se envían los datos de forma automática de Glimp a la aplicación web Nightscout donde son almacenados en la base de datos MongoDB como se muestra a continuación, en la imagen 80, en donde se puede observar que el valor se almacena de forma correcta.

Imagen 80. Valores almacenados de las lecturas del sensor.


```
_id: ObjectId("5f6a76ed63cc59a851bd2aeb")
date: 1600810110000
dateString: "2020-09-22T21:28:30.000Z"
rssi: 100
device: "glimp://libre/0M0060W2XTM"
direction: "Flat"
rawbg: 124
sgv: 124
type: "sgv"
utcOffset: -300
sysTime: "2020-09-22T21:28:30.000Z"
```


Fuente: Autor.

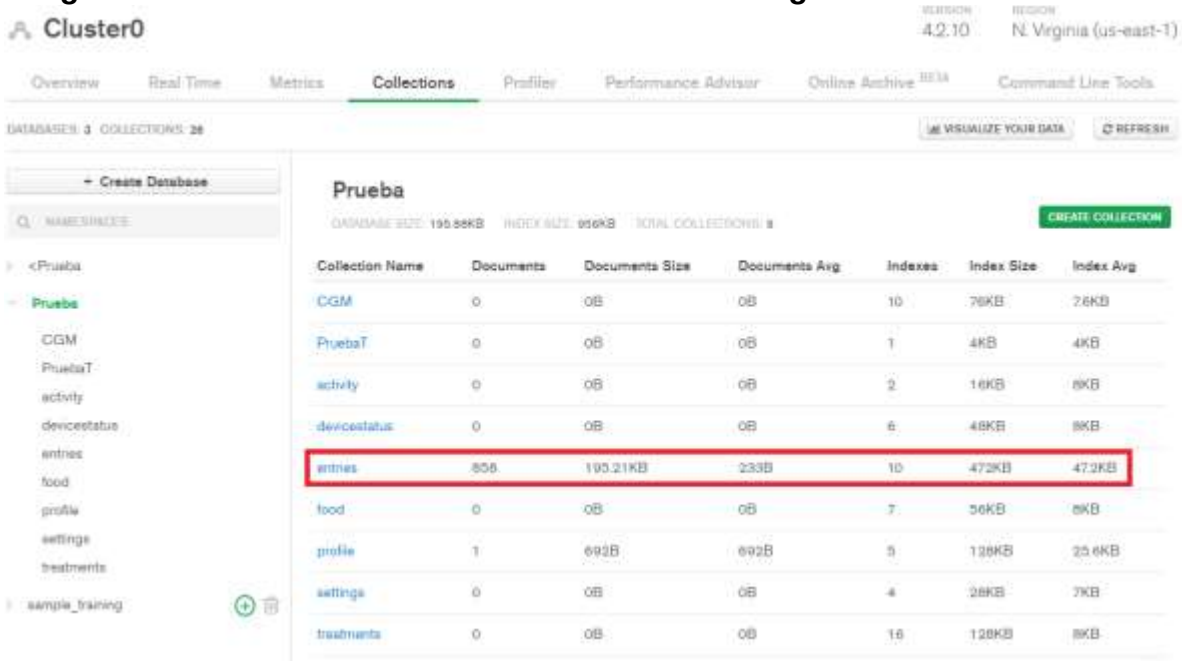
Una vez que se valida el almacenamiento correcto de los datos enviado de Glimp a la aplicación web Nightscout, se revisa los datos almacenado en la base de datos MySql con el software Workbench MySql, en la tabla “sensorFreeStyle” como se visualiza en la imagen 81, la cual está vacía para visualizar con más facilidad el traspaso de los datos de MongoDB a MySql.

Imagen 81.Resultados base de datos limpia.

Fuente: Autor

En esta etapa de validación se encontró que almacenados en la base de datos de MongoDB se encuentran exactamente 858 registros del sensor en la colección entries dentro de la base de datos “Prueba”, y se almacenaran en la base de datos generada automáticamente por ClearDB ya que se decide utilizar esta base de datos en la nube para las pruebas en el trabajo de grado. Los datos serán almacenados en la tabla SensorFreestyle como se mencionó anteriormente.

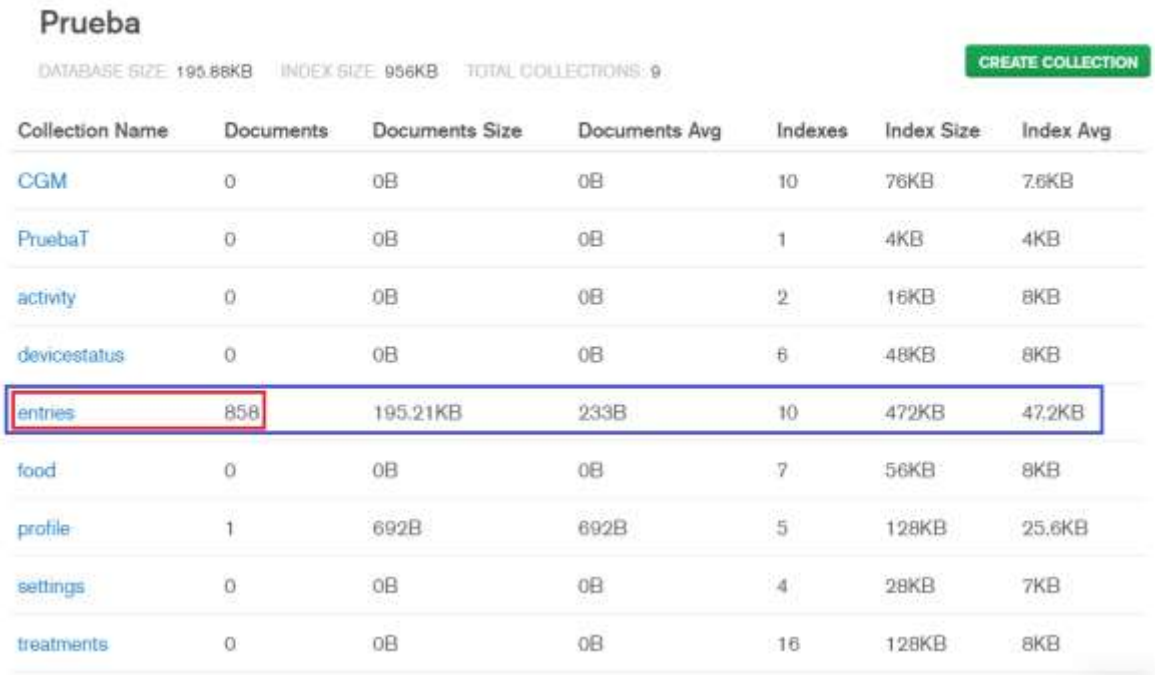
Imagen 82. Cantidad de datos almacenados en MongoDB.



Fuente: Autor.

En la imagen 82, se puede visualizar el Cluster que genera MongoDB por defecto y dentro de él, la base de datos que se crea en el Deploy de la aplicación web de Nightscout, llamada “Prueba” y dentro de la colección “entries” se encuentran los datos almacenados del sensor Freestyle libre, estos son leídos por la aplicación desarrollada en Flask, procesados y enviados al servidor de base de datos MySql.

Imagen 83. Cantidad de datos en la colección de interés de MongoDB.



Prueba

DATABASE SIZE: 195.88KB INDEX SIZE: 956KB TOTAL COLLECTIONS: 9

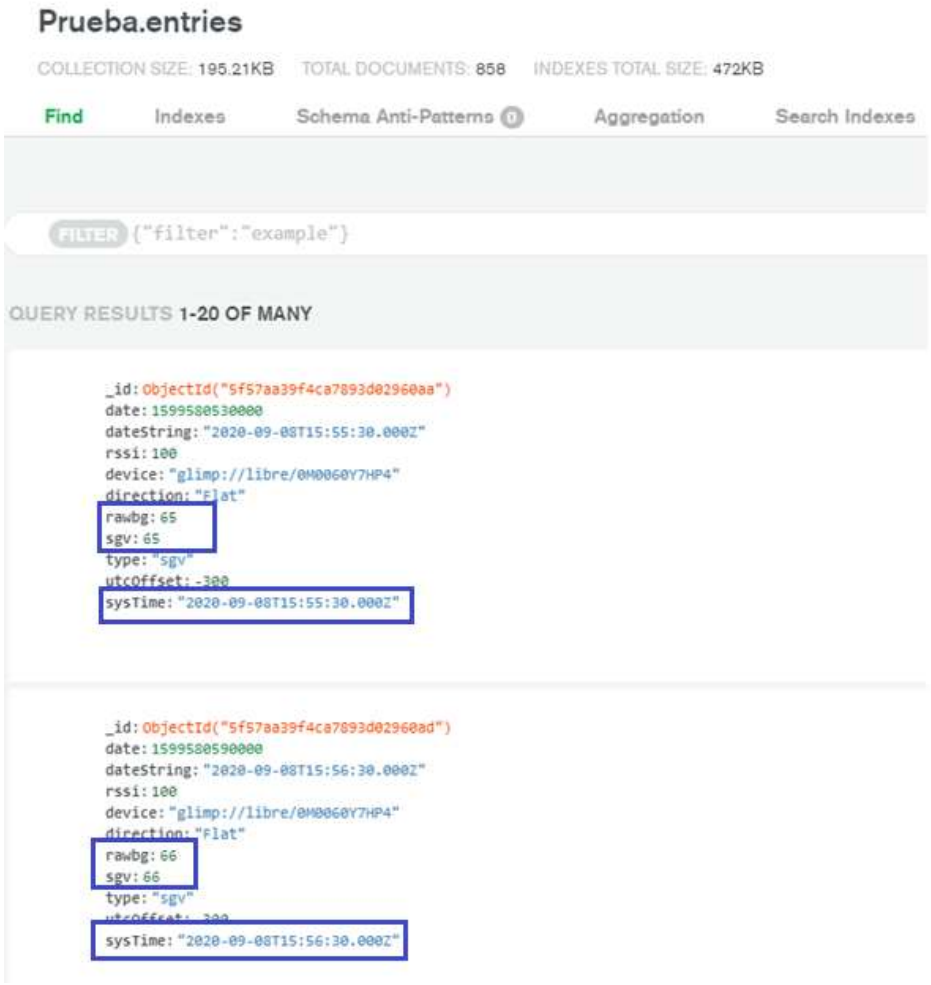
CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
CGM	0	0B	0B	10	76KB	7.6KB
PruebaT	0	0B	0B	1	4KB	4KB
activity	0	0B	0B	2	16KB	8KB
devicestatus	0	0B	0B	6	48KB	8KB
entries	858	195.21KB	233B	10	472KB	47.2KB
food	0	0B	0B	7	56KB	8KB
profile	1	692B	692B	5	128KB	25.6KB
settings	0	0B	0B	4	28KB	7KB
treatments	0	0B	0B	16	128KB	8KB

Fuente: Autor

Se puede señalar que en la imagen 83 se visualiza con mayor claridad los datos de interés, recordando que el total de datos a almacenar son 858 y que en la imagen 84 se encuentran los primeros datos dentro de collection entries.

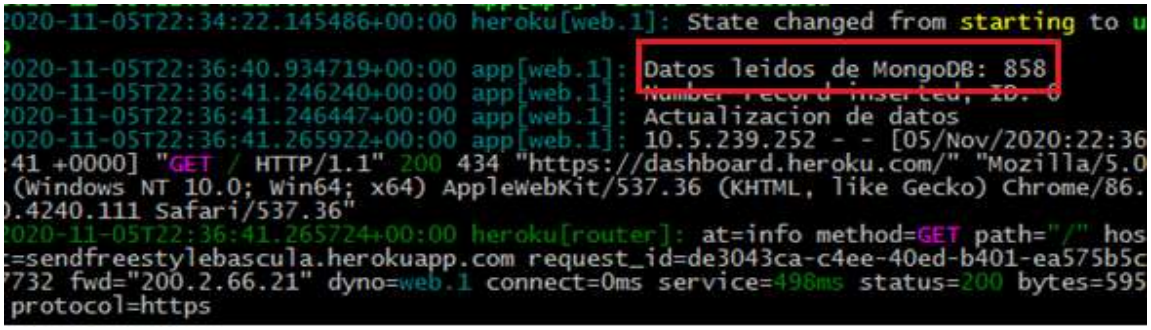
Imagen 84. Datos disponibles en collection entries.



Fuente: Autor

Al tomar como referencia los dos primeros datos para la validación de la transmisión y almacenamiento de los datos en la base de datos se busca evaluar la integridad de los datos en este proceso, por lo que se procede a visualizar en la cola de registros de Heroku, una que se desplegó la aplicación y se tiene en funcionamiento. Se registra en la cola de registros de Heroku los datos leídos, como se observa en la imagen 85 resaltado con un rectángulo rojo, en donde da aviso del total de datos leídos, 858, valor que es correcto teniendo en cuenta lo mostrado en la Imagen 84.

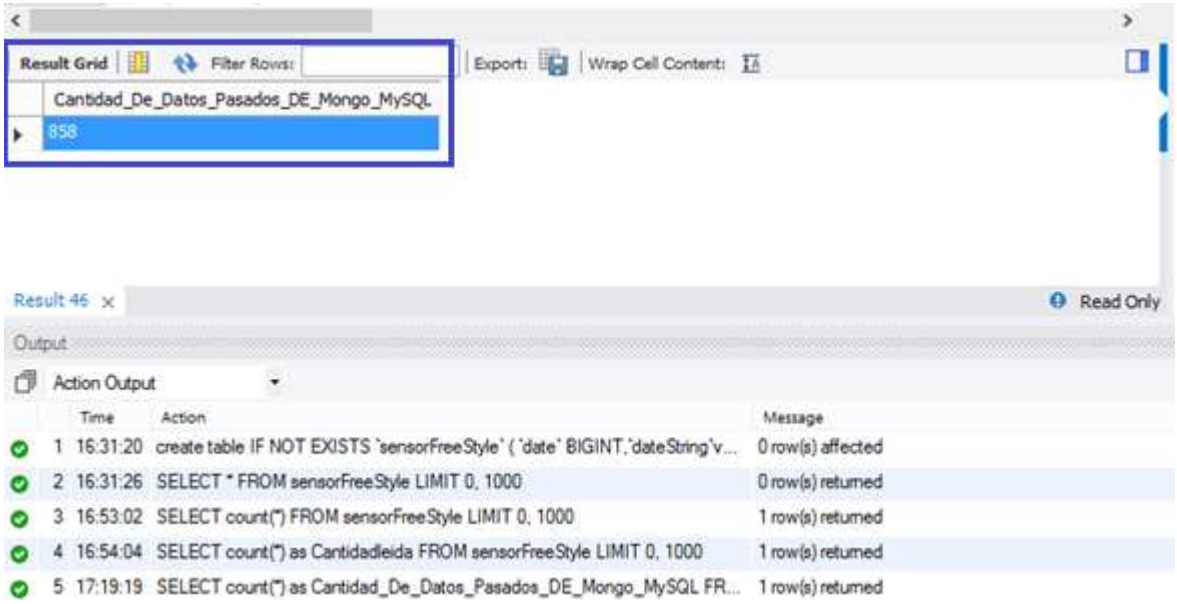
Imagen 85. Conteo, datos leídos desde la aplicación web con Flask en el servidor.



Fuente: Autor

Después de que la aplicación desarrollada en Flask lee los datos, se hace el proceso de trasmisión hacia el servidor MySQL, donde se almacenaran todos los valores de glucosa recogidos por el usuario. Para la visualización del total de los datos registrados en la base de datos MySQL se realiza una consulta nuevamente en el software WorkBench como se observa en la Imagen 86. Y resultado de la cantidad de datos almacenados es coherente con los que estaban en la base de datos MongoDB.

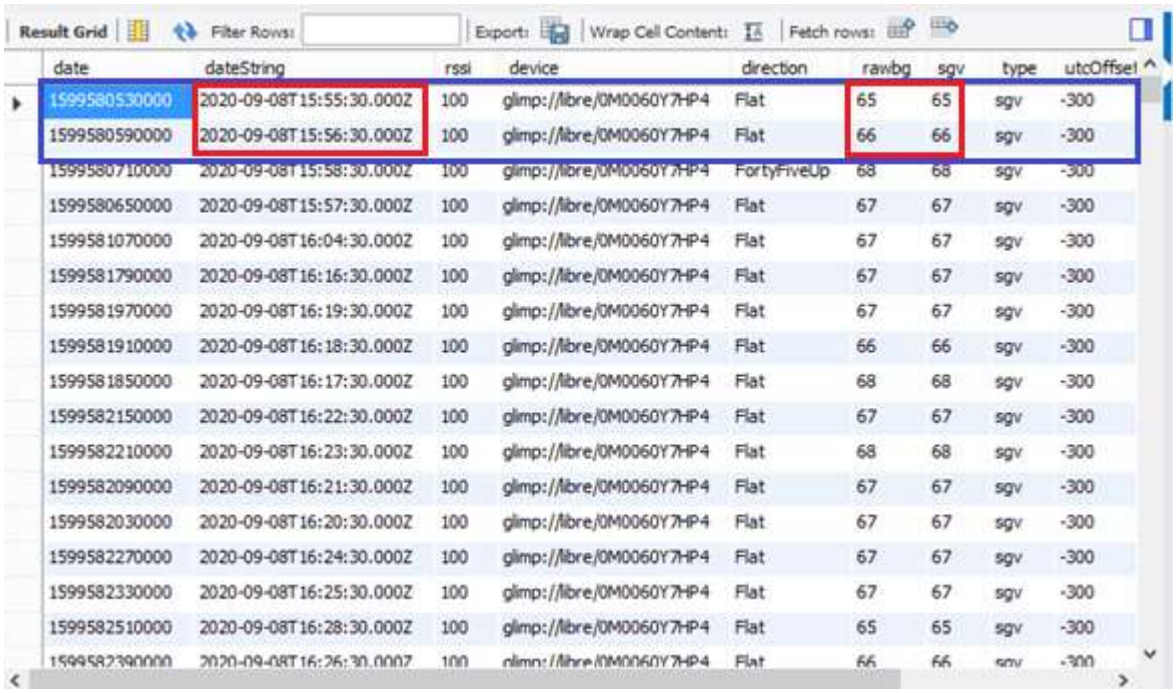
Imagen 86. Conteo datos insertados en tabla sensor Freestyle en base de datos MySql.



Fuente: Autor

Para realizar una validación de los datos se compara los datos de referencia almacenados en MongoDB que se encuentran en la imagen 84, con los que se tienen en la tabla sensorFreeStyle en la base de datos MySql (ver Imagen 87). El resultado de la comparación es que son exactamente iguales, conservando el orden de almacenamiento y los valores.

Imagen 87. Validación datos tabla Freestyle en WorkBench.



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

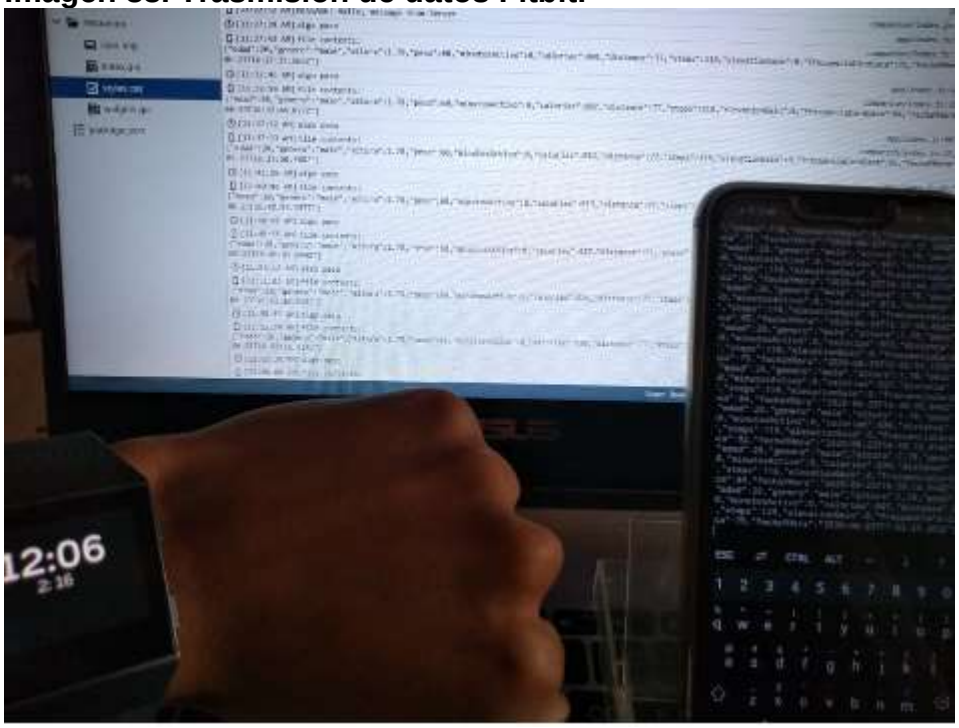
date	dateString	rssi	device	direction	rawbg	sgv	type	utcOffset
1599580530000	2020-09-08T15:55:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	65	65	sgv	-300
1599580590000	2020-09-08T15:56:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	66	66	sgv	-300
1599580710000	2020-09-08T15:58:30.000Z	100	glimp://libre/0M0060Y7HP4	FortyFiveUp	68	68	sgv	-300
1599580650000	2020-09-08T15:57:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599581070000	2020-09-08T16:04:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599581790000	2020-09-08T16:16:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599581970000	2020-09-08T16:19:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599581910000	2020-09-08T16:18:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	66	66	sgv	-300
1599581850000	2020-09-08T16:17:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	68	68	sgv	-300
1599582150000	2020-09-08T16:22:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599582210000	2020-09-08T16:23:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	68	68	sgv	-300
1599582090000	2020-09-08T16:21:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599582030000	2020-09-08T16:20:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599582270000	2020-09-08T16:24:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599582330000	2020-09-08T16:25:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	67	67	sgv	-300
1599582510000	2020-09-08T16:28:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	65	65	sgv	-300
1599582390000	2020-09-08T16:26:30.000Z	100	glimp://libre/0M0060Y7HP4	Flat	66	66	sgv	-300

Fuente: Autor

7.3 VALIDACIÓN FITBIT.

Para la validación se realiza un comparativo entre los datos obtenidos en el Fitbit y los datos almacenados en el servidor de base de datos MySQL. La transmisión de los datos del Fitbit se puede visualizar a través del sdk de Fitbit studio, y en el cli de Heroku. Un ejemplo de la visualización se encuentra en Imagen 88, donde el reloj Fitbit está ejecutando la aplicación que se encuentra en el sdk de Fitbit, y se sincroniza a el companion (dispositivo móvil) para realizar el envío hacia la base de datos.

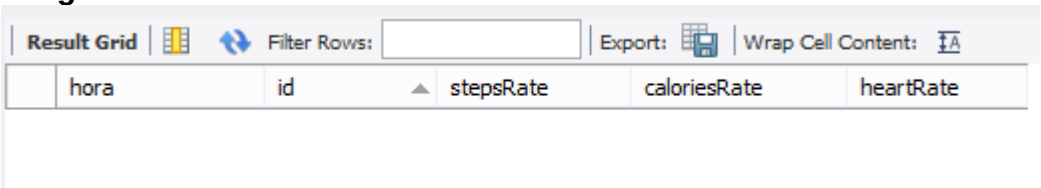
Imagen 88. Trasmisión de datos Fitbit.



Fuente: Autor.

Para poder validar que los datos están llegando correctamente a la base de datos, en el gestor de base de datos WorkBench se realiza el proceso para generar una tabla vacía en el servidor MySQL, encargada de recibir los datos del reloj Fitbit Ionic como se observa en la imagen 89.

Imagen 89. Tabla de datos Fitbit



Fuente: Autor.

A continuación, se muestra el envío de los datos almacenados localmente en el reloj en un lapso corto de tiempo, (ver Imagen 90). Al visualizar la imagen se observan los archivos Json que contiene el listado de llave-valor de los datos que se están enviando, donde las llaves que contienen los valores son: Hora, id, stepsRate, caloriesRate y heartRate, Estas llaves se visualizan dentro de los corchetes y lo que esta continuación después de los dos puntos es el valor que almacena esa llave o variable, esto es crucial para que el servidor que contiene la aplicación hecha en

Flask pueda entender y procesar lo que fue enviado por el método “POST”, generado por el companion.

Imagen 90. Visualización consola SDK, Datos enviados del reloj al servidor.

Fuente: Autor

Una vez que se ha realizado él envió correcto de la información, se procede a la visualización y validación de la información enviada por reloj Fitbit en la base de datos, revisando la tabla que se genera en la consulta SQL, en el servidor a través del gestor de base de datos WorkBench, se obtienen los resultados que se encuentran en la Imagen 91.

Imagen 91. Resultados de datos almacenado en la tabla fitbit en MySQL.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hora	id	stepsRate	caloriesRate	heartRate
2020-10-00 16:15:09	8L58W7	54783	12271	107
2020-10-00 16:15:09	8L58W7	54783	12271	108
2020-10-00 16:15:09	8L58W7	54783	12271	109
2020-10-00 16:15:09	8L58W7	54783	12271	109
2020-10-00 16:15:09	8L58W7	54783	12271	112
2020-10-00 16:15:09	8L58W7	54783	12271	113
2020-10-00 16:15:09	8L58W7	54783	12271	113

Fuente: Autor.

En la comparación entre las Imágenes 90 y 91. Se observa que todos los datos llegaron de forma correcta, es decir no sufrieron ninguna alteración en su almacenamiento brindando integridad el proceso de almacenamiento a los datos enviados por el reloj.

Tabla Costo Final de la Implementación

Tabla 8. Costos Final Implementación.

Elemento	Valor Final
Sensores Freestyle x 3	600.000
Tarjeta Arduino	85.000
Reglo Fitbit Ionic	700.000
HC-06	17.000
Licencia Windows	105.000
Licencia paquete office	285.000
Módulo HX711	25.000
Celdas de carga x 4	28.000
Valor Total (COP)	1’845.000

8. CONCLUSIONES.

Desafortunadamente en la actualidad el número de pacientes diabéticos identificados está en aumento y según la organización mundial de la salud, en el 2014 se tuvo 422 millones de adultos identificados con esta enfermedad en comparación con 1980 en donde la cantidad total fue de 108 millones de personas, variando el índice de prevalencia de 4,7% al 8,5 %. En Colombia para el año 2017 se tenía un índice de prevalencia de 8,36%, es decir 2.206.440 personas afectadas por esta enfermedad y estas cifras no dejan de aumentar.

El mayor peligro de la diabetes es que conforme va pasando el tiempo y esta no se controla correctamente va generando un deterioro en órganos como, los ojos, corazón, riñones, sistema nervioso, entre otros, generando que las personas diabéticas tengan que recurrir más o menos periódicamente a las salas de urgencias por síntomas tales como el pie diabético. Por lo que se requiere de herramientas que permitan el monitoreo de las variables fisiológicas de los pacientes diabéticos de forma continua.

Este trabajo de grado se realiza en conjunto con la investigación en desarrollo de algoritmos de predicción de insulina de la Universidad Complutense de Madrid, en el grupo de investigación, requiere la actualización de su base de datos MySQL con todas las variables fisiológicas pertinentes al campo de la diabetes (nivel de glucosa, frecuencia cardiaca, cantidad de pasos, cantidad de calorías, peso), de tal forma que se pueda actualizar los dato en tiempo real o en lapsos de 5 minutos aproximadamente hablando específicamente del Fitbit y del sensor Freestyle.

El sistema E-Health propuesto en este trabajo de grado tiene la restricción de que es necesario el acceso a Internet para poder subir los datos, en caso de no tener conexión los datos serán almacenados y posteriormente serán transmitidos cuando se cuente con la conexión a la red, internet.

Con el desarrollo del sistema de adquisición, procesamiento y almacenamiento de algunos de los datos fisiológicos relevantes (nivel de glucosa, frecuencia cardiaca, cantidad de pasos, calorías de calorías y el peso) para el cálculo de la predicción para el suministro de insulina en pacientes diabéticos se puede concluir que existen diversos dispositivos, herramientas para el seguimiento y control del nivel de glucosa, al igual que otras variables relevantes para los pacientes diabéticos, algunos son invasivos y otros como los que se utilizaron en trabajo de grado que son no invasivos.

Para el desarrollo de este sistema se optó por utilizar los sensores Freestyle Libre de Abbott, de fácil acceso en latino América y específicamente en Colombia y el reloj Fitbit Ionic también disponible en Colombia. Lo que permite al usuario realizar mediciones desde zonas urbanas o rurales, estas mediciones pueden almacenarse de forma local y en la nube siempre y cuando el usuario cuente con una conexión a

internet. Esta última opción, es recomendada dado que permite hacer un seguimiento de las variables sensadas por parte del personal médico.

En el mercado los sensores de glucosa Freestyle Libre de Abbott son de fácil acceso y su precio en comparación a la competencia son inferiores llegando a costar 400.000 pesos colombianos (2 parches) al mes teniendo en cuenta que la duración es de 14 días para cada parche, mientras que sus competidores como Dexcom G5/6 pueden llegar a costar más de un millón y medio de pesos al mes sin contar el tiempo y los costos de envió ya que estos tendrán que importarse desde España u otras sucursales. Por otro lado, existen los sensores suministrados por la empresa Mediatric, con el inconveniente de que ellos solo distribuyen estos productos a personas con órdenes médicas.

En cuanto la integridad de la información en la transmisión se concluye que debido a las tecnologías, protocolos y recursos que se utilizaron para la realización del trabajo de grado, se garantiza, como se muestra en la sección de validación, que la información no es modificada en su transmisión ni almacenamiento para ninguno de los tres dispositivos utilizados para la captación de la información (Bascula, reloj Fitbit Ionic y sensor Freestyle libre).

También se puede concluir que los sensores Freestyle Libre de Abbott medidores de glucosa, se sincronizan automáticamente con la aplicación web Nightscout una vez se tienen lecturas disponibles en el sensor. Por otro lado, el dispositivo Fitbit almacena los datos sensados (calorías, frecuencia cardiaca y cantidad de pasos) localmente, enviando estos datos al servidor cada 5 minutos (este tiempo es programable) si este tiene conexión a internet.

Por último, la báscula realiza él envió del valor de peso a través del dispositivo móvil, teniendo tiempos de envió de máximo dos minutos (ver tabla 7) para la recepción en la aplicación implementada en Heroku. La transmisión de estas variables permite realizar un control y monitoreo oportuno, con un mayor número de variables a considerar en el tratamiento del paciente diabético que le aportan mayor cantidad de información en un periodo de tiempo al profesional de la salud, en comparación con los mecanismos de control tradicional para la diabetes (ejemplo, los pacientes deben llevar un registro manual de las medidas de glucosa diariamente).
En cuanto a la validación de los datos adquiridos por la báscula, se puede concluir que la tecnología Bluetooth permite él envió confiable de datos al dispositivo móvil con línea de vista de hasta 10 metros y con obstáculos de hasta 5 metros.

Por otro lado, con el Sensor Freestyle libre versión 1, que corresponde al que se está vendiendo actualmente en Colombia, tiene una limitante y es que necesita de un dispositivo receptor que cuente con tecnología NFC, para poder realizar la lectura del sensor. Este inconveniente se soluciona con la versión 2 del sensor.

El Fitbit brinda seguridad en la transmisión de la información por el hecho de ser un dispositivo IoT, el fabricante diseñó su SDK para que los datos sean enviados mediante el dispositivo móvil, en caso de no tener conexión con el dispositivo móvil, el reloj almacena de forma local los datos. Este reloj tiene una capacidad de almacenamiento de máximo 15 MB incluyendo la aplicación.

La visualización de la información se realiza por medio de la aplicación desarrollada en Flask, encargada de consultar los datos ya almacenados en la base de datos y mostrar el historial de los datos. La aplicación es compatible con diferentes terminales (móviles y locales) ya que fue desarrollada con tecnologías orientadas a servicios web. La aplicación esta implementada en el servidor Heroku que ofrece servicio de hosting gratuito para aplicaciones web que no consuman muchos recursos del propio servidor al mes (superen las Gigas de almacenamiento de la aplicación), como es el caso de este trabajo de grado.

Ya para finalizar, se puede concluir que el desarrollo de este sistema E-Health, brinda una alternativa eficiente al control del tratamiento de pacientes diabéticos a través de la adquisición de variables fisiológicas relevantes tanto para el paciente como para el profesional de la salud, lo anterior mediante un almacenamiento constante y ordenado de la información, contando con una interfaz que permita visualizar la información contenida en la base de datos.

La transferencia de resultados se realizará por medio de un artículo, en el que contendrá la explicación necesaria del uso e implementación del sistema propuesto en el trabajo de grado, con el objetivo de aportar a la comunidad de diabéticos, y en específico a la comunidad de Nightscout, centrada en brindar una aplicación gratuita para los pacientes diabéticos.

9. TRABAJOS FUTUROS

Al finalizar la elaboración de este trabajo de grado con el fin de complementar lo planteado a lo largo del documento, se sugiere lo siguiente:

- Considerar la implementación de más sensores biomédicos relevantes para el control de pacientes diabéticos (presión arterial, diastólica y sistólica, oscilometría entre otros)

- Mejorar la precisión de la báscula para lograr un error menor en grandes pesos sensados.

- Elaborar un artículo que permita explicar el sistema E-Health propuesto en este trabajo de grado, bridando una herramienta que les permita a las personas diabéticas llevar un monitoreo más a gusto de sus valores de glucosa en sangre, así como de las otras variables fisiológicas en un solo lugar.

- Crear una base de datos con el gestor integrado de Python (SQLite), ya que este gestor está diseñado específicamente para el almacenamiento de datos de IoT, y brinda varias características clave del lenguaje estructurado de consultas (SQL).

- Realizar una interfaz de usuario con tecnologías visuales (CSS y JavaScript) que permitan al usuario un mayor entendimiento de sus datos.

- Implementación de un servicio de Login, registro y otras características que le brinden seguridad a la aplicación como una app_Secret.

- Realizar una aplicación multiplataforma.

10. BIBLIOGRAFÍA

"Diabetes" [Online] página OMS, 2018. [Consultado en 26-Mar-2020]. Disponible en internet: https://www.who.int/es/news-room/fact-sheets/detail/diabetes.

ANARTE, RUIZ DE ADANA, CARRERA, DOMIGUEZ LOPEZ, MACHADO, GONZALES MOLERO, CABALLERO, DE LA HIGUERA, GONZALES ROMERO, SANCHEZ, SORIGUER. Tratamiento de la diabetes mellitus (I). Av Diabetol., vol. 12, n°. 18, pp. 1001–1012.

APABLAZA, Pamela, SOTO, Néstor y CODNER, Ethel. De la bomba de insulina y el monitoreo continuo de glucosa al pancreas artificial. EN Rev. Med. Chil. Mayo, 2017 vol. 145, n°. 5.

AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube. [En línea]. [consultado 04 de mayo 2020]. Disponible en: https://aws.amazon.com/es/ec2/.

B. Feldman. "Freestyle (TM): A small-volume electrochemical glucose sensor for home blood glucose testing," Diabetes Technol. En: Rev. 2000. Ther. vol. 2, n°. 2, pp. 221–229. doi: 10.1089/15209150050025177.

BONDIA, VEHÍ, PALERM y HERRERO. El Páncreas Artificial: Control Automático de Infusión de Insulina en Diabetes Mellitus Tipo 1. En: Rev. Iberoam. Automática e Informática Ind. RIAI. 2010., vol. 7, no. 2.

C. MARTÍN. Guía para personas con diabetes tipo 1 y 2. Natl. Inst. Diabetes Dig. Kidney Dis. 2013, vol. 2.

C. Sainz de Rozas Aparicio Jara Gallardo Anciano and C. Académico, "Manejo de los dispositivos de insulina. Información al paciente" 2013.

CALUPIÑA Moya, Dayan Jasmin y GARCIA VÁSCONEZ,Andrea Patricia . Diseño, simulacion y comparacion de controladores clasicos y avanzados, aplicados al modelo de glocosa-insulina en el sistema de páncreas artificial para pacientes con diabetes tipo 1. 2018.

Cardona. Tratamiento de insulina. [En línea] Fundación para la diabetes, 2017 [Consultado 28 de marzo 2020]. Disponible en: https://www.fundaciondiabetes.org/infantil/181/tratamiento-de-insulina-ninos.

CEPEDA DÍEZ, Jose, MEIJOME SÁNCHEZ, Xosé, SANTILLÁN GARCÍA, Azucena. Innovaciones en salud y tecnologías: las cosas claras. En: EfermeriaCyL, 1967. vol. 2, no. 5, pp. 413–420.

Comportamiento de variables clínicas, antropométricas y de laboratorio en pacientes con síndrome metabólico. [En línea]. [consultado 07 de mayo 2020]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1727-897X2011000200004.
D. EVANS, David. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. 2011.

En la diabetes tipo1, el páncreas no produce insulina porque el mismo cuerpo ha destruido las células que secretan estas una hormona que se encarga de regular el azúcar en la sangre.

ESPINOSA ESQUIVEL, MOISES. “DISEÑO y CONSTRUCCIÓN DE UNA CELDA DE CARGA.” SAN NICOLÁS DE LOS GARZA, N. L: FACULTAD DE INGENIERIA MECANICA Y ELECTRICA.

FAUS, SANCHEZ POZO. Tratamiento, control y seguimiento fármacoterapéutico del paciente. En: Rev. Pharm Care Esp. 2001. no. 3, pp. 240–247.

Flask documentation. https://flask.palletsprojects.com/en/1.1.x/foreword/#configuration-and-conventions

Federation International Diabetes, “Peaje mundial de diabetes,”. [Online]. 2019 [Citado en 01-Mar-2020]. Disponible en internet: https://www.diabetesatlas.org/en/sections/worldwide-toll-of-diabetes.html.

Fitbit Charge 3 | Pulsera de salud y actividad física avanzada. [En línea]. [consultado 03 de mayo 2020]. Disponible en: https://www.fitbit.com/co/charge3.

FreeStyle Libre | Lector.” [En Línea]. [consultado 04 de mayo,2020]. Disponible en: https://www.freestyle.com.co/productos/freestyleLibre/lector.html.

Google Cloud IoT: servicios de Internet de las cosas totalmente gestionados. [En línea]. [consultado 04 de may 2020]. Disponible en: https://cloud.google.com/solutions/iot/.

Hernández Márquez, Felipe, Benjamín, Rodríguez Medina y Alfonso Torres Ríos. 12BC. “Puente de Wheatstone Modificado Utilizado En Una Comparación Internacional En El Valor de 1 GΩ.” Centro Nacional de Metrología Km 4,5 Carretera a Los Cués, 76246, Querétaro, México., octubre 24, 12BC. http://cenam.mx/simposio2008/sm_2008/memorias/M2/SM2008-M233-1178.pdf.

IDF Diabetes Atlas 9th edition 2019. [En línea]. [consultado en 03 marzo 2020]. 2019.Disponible en: https://diabetesatlas.org/en/.

International Diabetes Federation - Facts & figures. [En línea]. Cifras de estudios de Diabetes globales 2019. [Consultado en 03 marzo 2020]. Disponible en: https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html.

Introducción a las redes WiFi Materiales de entrenamiento para instructores de redes inalámbricas. [EN línea]. International Centre for Theoretical Physics 2012. Disponible en: https://docplayer.es/10897654-Introduccion-a-las-redes-wifi-materiales-de-entrenamiento-para-instructores-de-redes-inalambricas.html

J. A. Blaya, H. S. F. Fraser, and B. Holt. E-Health Technologies Show Promise In Developing Countries. Health Aff., vol. 29. no. 2. 2010 pp. 244–251.

LACÁMARA ORMAECHEA, Nerea, BALSEIRO CAMPOAMOR, Marina, RUIZ SERRANO, Aranzazu, ROYUELA, Ana y MARTINEZ BADAS, Itziar. Relacion entre calidad de vida y control metabólico, tipos de tratamiento con insulina y monitorización de glucemia en diabetes mellitus tipo 1. Volumen 10. 2da edición. España: 2019 P. 60-68.

LACÁMARA ORMAECHEA, Nerea, BALSEIRO CAMPOAMOR, Marina, RUIZ SERRANO, Aranzazu, ROYUELA, Ana y MARTINEZ BADAS, Itziar. Óp. Cit., p. 61.

León, Isaura. Guías para el diagnóstico y tratamiento el síndrome metabólico y la diabetes tipo 2 (DM2). Nuevos criterios. En: Salus. 2004 vol. 8, n°. 1. pp. 3–5.

LEVY, VIDAL y JANSÀ. Bombas de insulina. Una alternativa en el tratamiento de la DM1. En: Mesa redonda. Diabetes Mellitus. 2004. vol 60 n° 2, p 55-60.

MALDONADO PAZ, Gerardo y DE LA CRUZ BERNARDA TÉLLEZ ALANÍS, María. Diabetes Mellitus Tipo 2 y sus efectos sobre procesos de Cognición Social. Trabajo de grado Profesional para obtener el grado de Doctor en Psicología. Estado de Morelos: Universidad Autónoma, 2019. 123 p.

MEDIAVILLA BRAVO, José Javier. Complicaciones de la diabetes mellitus. Diagnóstico y tratamiento. En: Rev. SEMERGEN. 2001 vol. 27, pp. 132–145.
Medidores continuos de glucosa, ¿qué son? - Asociación Diabetes Madrid. [En línea]. [consultado 25 de marzo 2020]. Disponible en: https://diabetesmadrid.org/medidores-continuos-glucosa/.

MedlinePlus. Presión arterial alta. [En línea]. Medlineplus.gov. 2016. [consultado 01 de abil,2020]. Disponible en: https://medlineplus.gov/spanish/highbloodpressure.html.

Memorias organizacionales en la era del almacenamiento en la nube.” [En línea]. [consultado 04 de mayo,2020]. Disponible en: https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/6979/8659.

PRIETO DONATE, Francisc. 4. Tecnología Bluetooth 4.1. Introducción. [En línea]. [consultado 25 de septiembre 2018] Disponible en: http://bibing.us.es/proyectos/abreproy/40048/fichero/VOLUMEN+1.+MEMORIA%252F4.+Tecnolog% C3%ADa+Bluetooth.pdf

PRIETO DONATE, Francisco. GPRS (General Packet Radio Service). [En línea]. [consultado 25 de septiembre 2018]. Disponible en: http://bibing.us.es/proyectos/abreproy/11372/fichero/Memoria%252F03+-+GPRS.pdf

Prueba de hemoglobina A1c: Información en MedlinePlus sobre pruebas de laboratorio. [En línea]. [consultado 24 de marzo, 2020]. Disponible en: https://medlineplus.gov/spanish/pruebas-de-laboratorio/prueba-de-hemoglobina-a1c/.

Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure. [En línea]. [consultado 04 de mayo 2020]. Disponible en: https://azure.microsoft.com/es-es/overview/what-is-azure/#most-popular-questions.

Qué valores de hiperglucemia determinan la existencia de un día. [En línea]. [consultado 25 de marzo 2020]. Disponible: https://www.sanitas.es/sanitas/seguros/es/particulares/biblioteca-de-salud/diabetes/valores-hiperglucemia.html.

Rayman, Kröger, and Bolinder. Could FreeStyle LibreTM sensor glucose data support decisions for safe driving. En: Rev. Diabet. Med. Abril, 2018. vol. 35, no. 4, pp. 491–494.

RIVAS ALPIZAR, Elodia, ZERQUERA TRUJILLO, Gisela, Hernández Gutiérrez, Caridad y VICENTE SÁNCHEZ, Belkis. Manejo práctico del paciente con diabetes mellitus en la Atención Primaria de Salud. En: Finlay Rev. Enfermedades no Transm., vol. 1, no. 3, pp. 229–251, 2011.

RODRIGUEZ PALOMO, Inmaculada. Mejora del diseño de un prototipo de sensor no invasivo para la medida de glucosa en sangre.Trabajo de grado profesional Ingeniería Aeronáutica. Sevilla: Escuela Técnica Superior de Ingeniería Universidad de Sevilla.Dep. de Ingeniería de Sistemas y Automática. 2016

RODRÍGUEZ y MENDOZA. Factores de riesgo de diabetes mellitus tipo 2 en población adulta. Barranquilla, Colombia. En: Rev. Colomb. Endocrinol. Diabetes Metab. Mayo, 2019, vol. 6, no. 2, pp. 86–91.

ROMERO-VALENZUELA, Erika, ZONANA NACACH, Abraham y DE LOS ANGELES COLIN GARCIA, María. Control de glucosa en pacientes que asistieron

al programa de educación DiabetIMSS en Tecate, Baja California. En: Med Int Méx. 2014. p. 554-564.
S. N. Davis and G. Lastra-González. Diabetes y Bajo Nivel de Glucosa (Hipoglicemia). En: J. Clin. Endocrinol. Metab. Agosto, 2008., vol. 93, no. 8, pp. E1–E1.

SERRANO, Valentina, LARREA MANTILLA, Laura, RODRÍGUEZ GUTIÉRREZ, René, SPENCER BONILLA, Gabriela, MALAGA, German, HARGRAVES, Ian y MONTORI, Víctor. Toma de decisiones compartidas en la atención de pacientes con diabetes mellitus: Un desafío para Latinoamérica. Rev. Med. Chil. Mayo, 2017, vol. 145, n°. 5.

SERRANO, Valentina, LARREA MANTILLA, Laura, RODRÍGUEZ GUTIÉRREZ, René, SPENCER BONILLA, Gabriela, MALAGA, German, HARGRAVES, Ian y MONTORI,
VALDÉS PÉREZ, Fernando y PALLÁS ARENY, Ramón. Microcontroladores: fundamentos y aplicaciones con PIC. Marcombo Ediciones Tecnícas, 2007.

VARGAS URICOECHEA, Hernando y CASAS FIGUEROA, Luz. Epidemiology of diabetes mellitus in South America: The experience of Colombia. En: Clin. e Investig. en Arterioscler. 2016, vol. 28, n°. 5.

Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure) - Health Encyclopedia - University of Rochester Medical Center. U of R, 2016. [En línea]. Universidad de Rochester: centro médico. Disponible en: ttps://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P03963. [consultado 23 de abril 2020].

11. ANEXOS

Anexo A. Librerías utilizadas y asignación de pines para salida del módulo HX711.

```
#include "HX711.h"

// Pin de datos y de reloj
byte pinData = 3;
byte pinClk = 2;
```

Anexo B. Inicialización y Calibración Bascula.

```
HX711 bascula;

// Parámetro para calibrar el peso y el sensor
float factor_calibracion = 25880.0;
//Este valor del factor de calibración funciona para la bascula de estre tabajo.
//Inicial 20780.0 despues 25880.0 despues 28550.0 despues 27080.0, despues 25880.0
//(Mayor exactitud hasta el monento peso ligero)
void setup() {
  Serial.begin(9600);
  // Iniciar sensor
  bascula.begin(pinData, pinClk);
  // Aplicar la calibración
  bascula.set_scale();
  // Iniciar la tara
  // No tiene que haber nada sobre el peso
  bascula.tare();

  // Obtener una lectura de referencia
  long zero_factor = bascula.read_average();
  // Mostrar la primera desviación
  //Serial.print("Zero factor: ");
  //Serial.println(zero_factor);
}
```

Anexo C. Adquisición del valor de peso y transmisión por módulo HC–06.

```
void loop() {

  // Aplicar calibración
  bascula.set_scale(factor_calibracion);
  delay(1000);
  // Mostrar la información para ajustar el factor de calibración
  //Serial.print("Leyendo: ");
  float valBascula = bascula.get_units();
  if ( valBascula < 0.0  ){
      Serial.print(valBascula*-1, 1);
      delay(400);

    }
  if ( valBascula >= 0.0  ){
      Serial.print(valBascula, 1);
      delay(400);
    }

  // Obtener información desde el monitor serie
  /*if (Serial.available())
  {
    char temp = Serial.read();
    if (temp == '+')
      factor_calibracion += 10;
    else if (temp == '-')
      factor_calibracion -= 10;
  }*/
}
```

Anexo D. Contenido de carpeta APP del SDK de Fitbit.

```
// LIBRARIES

import clock from "clock"
import document from "document"
import { user } from "user-profile"
import { outbox } from "file-transfer";
import * as fs from "fs";
import * as messaging from "messaging";
import { me as appbit } from "appbit";
import { today } from "user-activity";
import { HeartRateSensor } from "heart-rate";
import * as dayToday from "../date/date"
import {  listDirSync, unlinkSync, writeFileSync, statSync } from "fs"
import { me } from "appbit";
import { BodyPresenceSensor } from "body-presence"
```

```
// INITIALIZATION OF VARIABLES
const myClock = document.getElementById("myClock")
const countdown = document.getElementById('countdown')
const wifi = document.getElementById('wifi')
const nowifi = document.getElementById('nowifi')
const sinchronization = document.getElementById('sinchronization')
const nosinchronization = document.getElementById('nosinchronization')
const steps = document.getElementById('steps')
const calories = document.getElementById('calories')
const heartRate = document.getElementById('heartRate')
const date = document.getElementById('date')
```

```
nowifi.style.visibility = 'hidden'
wifi.style.visibility = 'hidden'
sinchronization.style.visibility = 'hidden'
nosinchronization.style.visibility = 'hidden'


// DELETE FILES WHEN APP CLOSES
const dirIter
var contStart = 0
const listDir = listDirSync("/private/data");
while((dirIter = listDir.next()) && !dirIter.done) {
  unlinkSync(dirIter.value)
}
```

```
// EVENT PRESENCE SENSOR
if (BodyPresenceSensor) {
  const body = new BodyPresenceSensor();
  body.addEventListener("reading", () => {
    if (!body.present) {
      stop = 0 // NO ON WRIST
    } else {
      stop = 1 // ON WRIST
    }
  });
  body.start();
}
// CLOCK AND COUNTDOWN
function startClock(){
  let today =  new Date()
  let day = today.getDay()
  let year = today.getFullYear()
  let month = today.getMonth()
  date.text = `${dayToday.day(day)}, ${today.getDate()} ${dayToday.month(month)} ${year}`

  clock.granularity = 'seconds'; // seconds, minutes, hours
  clock.ontick = function(evt) {
  myClock.text = ("0" + evt.date.getHours()).slice(-2) + ":" +
                        ("0" + evt.date.getMinutes()).slice(-2)
  }
}
function startCountdown() {
  min = parseInt(timer/60)
  sec = parseInt(timer%60)
  if(timer < 1){

    sychronizationConnection()  // it can send messages
    tryToSendAllInformation()
    takeData()

    timer = 1 * 10
    min = 5
    sec = 0
  }
  if(stop){
    countdown.text = `${min}: ${sec}`
    timer--
  }
  setTimeout(function() {
             startCountdown()
             }, 1000)
}
```

```
// SINCRONIZATION AND CONNECTION

function sychronizationConnection() {
  if (messaging.peerSocket.readyState === messaging.peerSocket.OPEN){
    nosinchronization.style.visibility = 'hidden'
    sinchronization.style.visibility = 'visible'
    storeDataSynchronization = 1 //sinchronization
    internetConnection()
  }
  if (messaging.peerSocket.readyState === messaging.peerSocket.CLOSED) {
  // Display error message
    sinchronization.style.visibility = 'hidden'
    nosinchronization.style.visibility = 'visible'
    wifi.style.visibility = 'hidden'
    nowifi.style.visibility = 'visible'
    storeDataSynchronization = 0 //no synchronization
  }
}

function internetConnection() {
  let internet = "INTERNET";
  writeFileSync("internetTest.txt", internet, "ascii");
    outbox
  .enqueueFile('internetTest.txt')
  .then((ft) => {
     console.log(`Transfer of ${ft.name} successfully queued.`);
  })
  .catch((error) => {
     console.log(`Failed to schedule transfer: ${error}`);
  })

  unlinkSync('internetTest.txt')
}

function sendMessage() {
  if (messaging.peerSocket.readyState === messaging.peerSocket.OPEN) {
    // Send the data to peer as a message
    console.log('se envio mensaje')
    messaging.peerSocket.send(internet);
  }
}
```

```
// MESSAGES SENT FROM COMPANION AND VERIFICATION OF CONNECTION
messaging.peerSocket.onmessage = function(evt) {
     verifyConnection(evt.data)
}
function verifyConnection(internetConnection) {
  if(internetConnection[0] === '1'){ // internet
    idUser = internetConnection[1]
    //console.log(`-------------------${internetConnection[1]}-----------------------`)
    nowifi.style.visibility = 'hidden'
    wifi.style.visibility = 'visible'
    storeDataInternet = 1
  }else {
    wifi.style.visibility = 'hidden'
    nowifi.style.visibility = 'visible'
    storeDataInternet = 0 //no internet
  }
}
// IF THERE IS INFORMATION STORED, IT TRY TO SEND ALL
function tryToSendAllInformation() {
  /*
  storeNumber -> count how many files there are stored, if this is higher than 0
  and bands storeDataInternet and storeDataSynchronization are on
  */
  if(storeNumber > 0 && storeDataInternet === 1 && storeDataSynchronization === 1){ //send information s
    sendDataStore()
  }
}
// SEND ALL FILES STORED
function sendDataStore() {
  for( let i = 0 ; i < storeNumber; i++){
    outbox
     .enqueueFile(`json${i}.txt`)
     .then((ft) => {
        console.log(`Transfer of ${ft.name} successfully queued.`);
     })
      .catch((error) => {
        console.log(`Failed to schedule transfer: ${error}`);
     })
  }
 deleteDataStore()
}
//DELETE ALL FILES SENT
function deleteDataStore() {
  for(let i = 0 ; i < storeNumber ; i++){
      unlinkSync(`json${i}.txt`)
  }
  storeNumber = 0
}
```

```
// SEND FILES BECAUSE THERE IS CONNECTION AND SYNCHRONIZATION
// IT ISN'T NECESSARY SAVE INFORMATION

function takeData() {

  let fecha = new Date()
  let today = `${fecha.getFullYear()}-${fecha.getMonth()}-${fecha.getDay()} ${fecha.getHours()}:${fecha.getMinutes()}:${fecha.getSecon

  person['hora'] = today
  person['id'] = idUser
  rateSteps(person)
}

//INFORMATION
function rateSteps(person) {
  let stepsRate
  if (appbit.permissions.granted("access_activity")) {
    person['stepsRate'] =  today.adjusted.steps
  }
  rateCalories(person)
}

function rateCalories (person) {
  let caloriesRate
  if (appbit.permissions.granted("access_activity")) {
    person["caloriesRate"] = today.adjusted.calories
  }
  heartRatePerson(person)
}
function heartRatePerson(person){
  if (HeartRateSensor) {
      const hrm = new HeartRateSensor();
      hrm.addEventListener("reading", () => {
      person['heartRate'] = hrm.heartRate
      //console.log(JSON.stringify(person))
      saveData(person)
    });
    hrm.start();
  }
}
```

```
// SAVE INFORMATION IN JSON FILE
function saveData(json_data) {
  if (storeDataInternet === 1 && storeDataSynchronization === 1){
    fs.writeFileSync('json.txt', json_data, "json");
    sendData()
  }else {
    // CYCLE WHERE THERE AREN´T CONNECTION AND SYNCHRONIZATION
    fs.writeFileSync(`json${storeNumber}.txt`, json_data, "json");
    storeNumber++
  }
}

// SEND INFORMATION DIDN'T SAVE
function sendData(){ //if there's a error with internet companion couldn't send anything, return
    outbox
    .enqueueFile('json.txt')
    .then((ft) => {
       console.log(`Transfer of ${ft.name} successfully queued.`);
    })
    .catch((error) => {
       console.log(`Failed to schedule transfer: ${error}`);
    })
    unlinkSync('json.txt') //delete basic json file
}
//START :)
startCountdown()
startClock()
```

Anexo E. Contenido de carpeta Companion del SDK de Fitbit.

```
import { inbox } from "file-transfer";
import { me } from "companion";
import * as messaging from "messaging";
import { settingsStorage } from "settings";
```

```
async function processAllFiles() {
  let file;
  while ((file = await inbox.pop())) {
    const payload = await file.text();
    //console.log(`file contents: ${payload}`);
    sendData(payload)
  }
}
inbox.addEventListener("newfile", processAllFiles);
processAllFiles()
function sendData(message) {
  console.log(`la información es ${message}`)
  if(message === 'INTERNET'){
    restoreSettings();
    //try internet connection
  }else {
    //send information
    saveInformation(message)
  }
}
function getData(accessToken)  {
  let date = new Date();
  let todayDate = `${date.getFullYear()}-${date.getMonth() + 1}-${date.getDate()}`; //YYYY-MM-DD

  fetch('https://api.fitbit.com/1/user/-/profile.json', {
    method: "GET",
    headers: {
      "Authorization": `Bearer ${accessToken}`
    }
  })
  .then(function(res) {
    return res.json()
  })
  .then(function(data) {
    //console.log(data.user.encodedId)
    if (messaging.peerSocket.readyState === messaging.peerSocket.OPEN) {
      let internetObject = ['1',data.user.encodedId]
      messaging.peerSocket.send(internetObject);
    }
  })
  .catch(err => {
    console.log('[FETCH]: ' + err);
    if (messaging.peerSocket.readyState === messaging.peerSocket.OPEN) {
       messaging.peerSocket.send('0');
    }
  })
}
```

```
// A user changes Settings
settingsStorage.onchange = evt => {
  if (evt.key === "oauth") {
    // Settings page sent us an oAuth token
    let data = JSON.parse(evt.newValue);
    getData(data.access_token) ;
  }
};
// Restore previously saved settings and send to the device
function restoreSettings() {
  for (let index = 0; index < settingsStorage.length; index++) {
    let key = settingsStorage.key(index);
    if (key && key === "oauth") {
      // We already have an oauth token
      let data = JSON.parse(settingsStorage.getItem(key))
      getData(data.access_token);
    }
  }
}
async function saveInformation(message) {
  try {
    let response = await fetch('https://sendfreestylebascula.herokuapp.com/fitbit', {
//    let response = await fetch('https://taller05-97e3f.firebaseio.com/TALLER05/Usuarios/registros2.j
      method: 'POST',
      headers: {
        'Content-Type': 'application/json;charset=utf-8'
      },
      body: JSON.stringify(message),
    });
    console.log("Prueba Json:")
    console.log(JSON.stringify(message));
    let result = await response.json();
    //console.log(result.message);
  }catch(err) {
    console.log(`ocurrio un error en el punto donde se enviarón los datos:  ${err}`)
  }
}
```

Anexo F. Archivo principal app.py de la aplicación hecha en Flask (Python).

```
from flask import Flask, render_template, request
from flask_pymongo import PyMongo
from flaskext.mysql import MySQL
import json
import time
import threading , sys

app = Flask(__name__)
app.config['MONGO_URI'] = 'mongodb+srv://Prueba:prueba@cluster0.xv7cj.mong
odb.net/Prueba?retryWrites=true&w=majority'
mongo = PyMongo(app)
mysql = MySQL()

app.config['MYSQL_DATABASE_USER'] = 'bdb20eaa9eef3c'
app.config['MYSQL_DATABASE_PASSWORD'] ='d0387f2a'
app.config['MYSQL_DATABASE_DB'] = 'heroku_42078ca09a517b2'
```

```
app.config['MYSQL_DATABASE_HOST'] = 'us-cdbr-east-02.cleardb.com'
mysql.init_app(app)

@app.route('/')
def home():
    countSensor = coutData("sensorFreeStyle")
    countPeso = coutData("pesousuarios")
    countFitbit = coutData("fitbit")
    print("Bienvenido")
    return render_template("index.html", sensor = countSensor, peso = coun
tPeso, fitbit = countFitbit)

@app.route('/bascula',methods = ['POST','GET'])
def bascula():
    if request.method ==  "POST":
        iduser = request.form['idUser']
        print(iduser)
        varpeso = (request.form['peso'])
        peso = float(varpeso)
        db2 =  mysql.connect()
        mycursor = db2.cursor()
        query = "INSERT INTO pesoUsuarios (iduser, peso) VALUES (%s,%s)"
        # query = "INSERT INTO pesoUsuarios (peso) VALUES (%s)"
        arrayQuery = ( iduser, peso )
        # print(query % (iduser,peso))
        print("entro querry:" +query%arrayQuery)
        try:
            mycursor.execute(query,arrayQuery)
            # print("Rowcont:  "+ mycursor.rowcount)
            db2.commit()
        except Exception as e :
            print("Eror: "+ e )
        db2.close()
        return render_template('index.html')
    return render_template('bascula.html')
# query = "INSERT INTO pesoUsuarios (iduser, peso) VALUES (%s,%s)"
            # arrayQuerry = tuple( iduser, peso )
            # mycursor.execute(querry,arrayQuerry)

#-------------------------------------------------------------------
@app.route('/peso')
# @app.route('/index',methods = ['POST','GET'])
def peso():
```

```
    read = readTables("pesousuarios")
    return render_template("peso.html", read = read)
@app.route('/sensor')
# @app.route('/index',methods = ['POST','GET'])
def sensor():
    read = readTables("sensorFreeStyle")
    return render_template("sensor.html", read = read )
@app.route('/graficainsulina')
# @app.route('/index',methods = ['POST','GET'])
def graficainsulina():
    read = readLastData("sensorFreeStyle","dateString")
    (date,valGlucosa) = ordenarGrafica(read, 1 , 5 )
    return render_template("graficainsulina.html",dateGlucosa = date, gluc
osa = valGlucosa)
@app.route('/graficapeso')
# @app.route('/index',methods = ['POST','GET'])
def graficapeso():
    read = readLastData("pesousuarios","fecha")
    (date,valPeso) = ordenarGrafica( read, 2, 1)
    return render_template("graficapeso.html",datePeso = date, peso = valP
eso)
@app.route('/graficacircular')
def graficacircular():
    read = readTables("sensorFreeStyle")
    zonasGraficaC = datacircular(read,5, 80, 120)
    return render_template("graficacircular.html",zonasGraficaC = zonasGra
ficaC)
@app.route('/graficacircularpeso')
def graficacircularpeso():
    read = readTables("pesousuarios")
    zonasGraficaC = datacircular(read,1,60,80)
    return render_template("graficacircularpeso.html",zonasGraficaC = zona
sGraficaC)

@app.route('/fitbit',methods = ['POST','GET'])
def fitbit():
    response = request.get_json()
    print(response)
    fecha = response['hora']
    caloriesRate = response['caloriesRate']
    heartRate = response['heartRate']
    stepsRate = response['stepsRate']
```

```
    print( "recived Data:  " + fecha + " " + str(stepsRate) + " " + str(ca
loriesRate) + " " + str(heartRate) )
    # ////////////////////////////////////////////////////////////////////
/////
    # --------------     Insersión MySQL  -------------------------------
-----
    db2 =  mysql.connect()
    mycursor = db2.cursor()
    datos= (fecha, stepsRate, caloriesRate, heartRate)
    querry = "INSERT INTO fitbit (fecha,stepsRate,caloriesRate,heartRate)
VALUES (%s,%s,%s,%s)"
    try:
        mycursor.execute(querry,datos)
        db2.commit()
        print("Number record inserted, ID:", mycursor.lastrowid)
    except:
        print("No Insersión ")
    db2.close()
        #--------------  Fin Insersión MySQL  ---------------------------
---------
    return '{"status":"funciona"}'

@app.route('/relojfitbit')
def relojfitbit():
    read = readTables("fitbit")                                  # Lectura
tabla fitbit (SELECT * FROM FITBIT)
    print(read[0])
    return render_template("relojfitbit.html", read = read )    # Mostrar
html relojfitbit al usuario y envio de los datos leidos

#-----------------------------------------------------------------
def coutData(tabla):
    db2 =  mysql.connect()
    mycursor = db2.cursor()
    querry = "select count(*) from "+ tabla
    error = ""
    try:
        mycursor.execute(querry)
        countData = mycursor.fetchone()
        print(countData[0])
        print("count row :", mycursor.rowcount)
        db2.commit()
        db2.close()
```

```
        return countData[0]
    except:
        print("Eror: "+ error)
        db2.close()
        countData = "vacio"
        return countData
    print(" ")

def readTables(tabla):
    db2 =  mysql.connect()
    mycursor = db2.cursor()
    querry = "select * from " + tabla
    error = ""
    try:
        mycursor.execute(querry)
        readData = mycursor.fetchall()
        print("count row :", mycursor.rowcount)
        db2.commit()
        db2.close()
        return readData
    except:
        print("Eror: "+ error)
        db2.close()
        readData = "vacio"
        return readData
def datacircular(tuplaMySQL, indice, bajaNormal, normalAlta):
    zonasDataUser =[0,0,0]
    for valData in tuplaMySQL:
        if valData[indice] >= bajaNormal and valData[indice] <= normalAlta
 :
            zonasDataUser[1] += 1#   Zona media (Nivel sanoooo)
        elif valData[indice] > normalAlta:
            zonasDataUser[2] += 1#   Zona Alta (Hiperglucemico)
        elif valData[indice] < bajaNormal:
            zonasDataUser[0] += 1#   Zona Baja (Hipoglucemico)
        # print("0-"+ str(zonasDataUser[0]) + " 1-
"+ str(zonasDataUser[1]) + " 2-"+ str(zonasDataUser[2]))
    return zonasDataUser
def readLastData(tabla,campo):
    db2 =  mysql.connect()
    mycursor = db2.cursor()
    lecturaDesc = "order by "+ campo +" desc limit 7"
    querry = "select * from " + tabla +" "+ lecturaDesc
```

```
    error = ""
    try:
        mycursor.execute(querry)
        readData = mycursor.fetchall()
        print("count row :", mycursor.rowcount)
        db2.commit()
        db2.close()
        return readData
    except:
        print("Eror: "+ error)
        db2.close()
        readData = "vacio"
        return readData
#   Return (Date, Val peso)
def ordenarGrafica(tuplaMySQL,indDate,indVal):
    dataVal     = []
    dataDate    = []
    print(tuplaMySQL)
    for dataTable in tuplaMySQL:
        dataDate.append(str(dataTable[indDate]))
        dataVal.append(int(dataTable[indVal]))
    return (dataDate, dataVal)

if __name__ == "__main__":
    app.run(debug=1)
```

Anexo G. Archivo server.py de la aplicación hecha en Python específicamente para lectura de sensor Freestyle libre.

```
import mysql.connector
import pymongo
import ssl


client = pymongo.MongoClient("mongodb+srv://Prueba:prueba@cluster0.xv7cj.mongodb.net/Prueba?retryWrites=true&w=majority",ssl=True ,ssl_cert_reqs = ssl.CERT_NONE)

# conexión con la base de datos
db = client['Prueba']
collection = db['entries']
```

```
lista = []
oldList = []
cont = 0
for valFreestyle in collection.find():
    #   Adqisicion datos almacenados
    oldList.append(valFreestyle)
    #   Fin
    valFreestyle.pop('_id')
    valFreestyle['date'] = int(valFreestyle['date'])
    listaAux = valFreestyle.values()
    listas = tuple(listaAux)
    lista.append(listas)
    cont+=1
print(cont)
#--------------     Insersión MySQL  ------------------------------------
db2 = mysql.connector.connect(
  host="us-cdbr-east-02.cleardb.com",
  user="bdb20eaa9eef3c",
  password="d0387f2a",
  database="heroku_42078ca09a517b2"
)

mycursor = db2.cursor()
querry = "INSERT INTO sensorFreeStyle (date,dateString,rssi,device,directi
on,rawbg,sgv,type,utcOffset,sysTime) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s
)"
print(querry%lista[0])
try:
    mycursor.executemany(querry,lista)
    db2.commit()
    print("Number record inserted, ID:", mycursor.lastrowid)
    db2.close()
except:
    print("No Insersión ")
```